Device Characteristics-based Differentiated Energy-efficient Adaptive Solution for Multimedia Delivery over Heterogeneous Wireless Networks

Ruiqi Ding

A Dissertation submitted in fulfilment of the

requirements for the award of

Doctor of Philosophy (Ph.D.)

to the

Dublin City University

Faculty of Engineering and Computing

School of Electronic Engineering

Supervisor: Dr. Gabriel-Miro Muntean

January 2014

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed:

ID No.:

Date:

Contents

List of Publication **viii**

Abstract **ix**

Acknowledgements **x**

List of Tables **xii**

List of Figures **xiv**

Abbreviations **xviii**

1 Introduction **1**

1.1 Research Motivation 1

1.2 Problem Statement 4

1.3 Solution Overview and Contributions 5

1.3.1 Energy-oriented Application-based System Profiling - ASP . . 7

1.3.2 Context-aware Differentiated Adaptive Routing - AWERA . . 8

1.3.3 Device-Differentiated Adaptive Content Delivery - DEAS . . 8

1.3.4 Energy-efficient Device-differentiated Cooperative Adaptive Multimedia Delivery - EDCAM 9

1.4 Thesis Structure 10

2 Background and Related Works **12**

2.1 Layered Protocols Model 12

2.2 Energy-Efficient Network Layer 14

2.2.1 Energy-aware Routing 16

2.2.2 Hierarchical Routing 18

2.2.3 Differentiate Devices 19

2.2.4 Geographic Routing 20

2.2.5 Energy Aware Routing with QoS Support 22

2.2.6 Content-based Routing 25

2.3 Cross-layer Approaches to Energy Efficiency 30

2.3.1 General Aproaches 32

2.3.2 Content Sharing 34

2.3.3 Traffic Shaping 35

2.3.4 Joint Routing and Sleep Scheduling 37

2.3.5 QoS Through Packet Prioritisation 40

2.4 Energy Efficient Mobile Platforms 44

2.4.1 Energy Efficient Wireless Interfaces 45

2.4.2 Processors 46

2.4.3 Graphical Display 48

2.4.4 Power Management at Operating System Level 51

2.4.5 Background Works 56

2.4.5.1 Measurement-based Energy Characterisation 56

2.4.5.2 Utilisation-based Energy Modelling 57

2.4.6 Energy Efficient Applications 60

2.4.6.1 Data Compression Techniques 61

2.4.6.2 Adaptive Control at Application Layer 64

2.4.6.3 Partial Caching 67

2.5 Chapter Summary 72

3 Energy-oriented Application-based System Profiling and Prototype-based Testing 73

3.1 Overview . . . 73
3.2 Contributions and Advantages . . . 75
3.3 ASP Architecture . . . 77
3.4 ASP Algorithm . . . 78
3.4.1 Overview . . . 78
3.4.2 Initialization Phase . . . 80
3.4.2.1 Regression Algorithms . . . 82
3.4.3 Monitoring Phase . . . 85
3.4.4 Working Phase . . . 86
3.5 Software Implementation . . . 88
3.6 Predictor Selection for Regression-based Modelling . . . 89
3.6.1 Real Tests . . . 89
3.6.2 Observations . . . 98
3.7 Regression-based Model Training . . . 99
3.7.1 Training Strategy . . . 100
3.8 Results of Model Training and Model Validation . . . 101
3.8.1 Prediction Using Current Value v.s. Using Calculated Power Value . . . 101
3.8.2 Training Using Predefined Tasks v.s. Incremental Adaptive Training Using Daily Used Applications . . . 103
3.8.3 Model Validation with Video Delivery . . . 103
3.8.4 Utilisation-based Estimation v.s. Power Signature-based Estimation . . . 109
3.8.5 Overhead Analysis . . . 110
3.9 Chapter Summary . . . 111

4 Application-aware Energy-efficient Routing Architecture, Algorithms and Testing 112

4.1 Overview . . . 112

4.2 AWERA Use Case Scenario 114
4.3 AWERA Architecture 116
4.4 AWERA Algorithm 120
4.4.1 Node Cost Computation Based on Application Profile 120
4.4.2 Routing Mechanism 122
4.4.3 Routing Cost Computation 123
4.5 Simulation Testing and Result Analysis - AWERA 125
4.5.1 Simulation Testing Environment 125
4.5.1.1 Brief introduction to NS-2 125
4.5.1.2 AWERA NS-2 Model 126
4.5.2 Simulation Scenario 1 - Power Source and Application Type Differentiation 127
4.5.2.1 Test Scenario Goals 127
4.5.2.2 Simulation Settings 128
4.5.2.3 Simulation Results 129
4.5.3 Simulation Scenario 2 - Light to Moderate Loaded Network . 134
4.5.3.1 Test Scenario Goals 134
4.5.3.2 Simulation Settings 134
4.5.3.3 Simulation Results 136
4.5.4 Simulation Scenario 3 - Heavy Loaded Network 141
4.5.4.1 Test Scenario Goals 141
4.5.4.2 Simulation Settings 141
4.5.4.3 Simulation Results 143
4.6 Chapter Summary 147

5 Device-differentiated Adaptive Content Delivery Architecture, Algorithms and Testing 148

5.1 Overview 148
5.2 DEAS Use Case Scenario 150

5.3 DEAS Architecture 151
5.4 DEAS Algorithm 152
5.5 Simulation Testing and Result Analysis - DEAS 155
5.5.1 Simulation Testing Environment 155
5.5.1.1 Brief introduction to NS-3 155
5.5.1.2 DEAS NS-3 Model 156
5.5.2 Simulation Scenario 1 - Effect of Background Traffic Increase in A WiFi Network 158
5.5.2.1 Simulation Goals 158
5.5.2.2 Simulation Settings 158
5.5.2.3 Simulation Results 161
5.5.3 Simulation Scenario 2 - Effect of Background Traffic in a LTE Network 163
5.5.3.1 Simulation Goals 163
5.5.3.2 Simulation Settings 163
5.5.3.3 Simulation Results 165
5.6 Chapter Summary 168

6 Energy-efficient Device-differentiated Cooperative Adaptive Multimedia Delivery Architecture, Algorithms and Testing 170
6.1 Overview 170
6.1.1 Hybrid Content Sharing and Quality Adaptation for Energy Saving at Client Device 171
6.1.2 Device Profiling for Quality Adaptation and Partner Selection of Content Sharing 173
6.2 EDCAM Use Case Scenario 175
6.3 EDCAM Architecture 176
6.3.1 Energy-oriented Application-based System Profiling 178
6.3.2 Energy Efficient Quality Adaptation 178

6.3.3 Partner Selection in Cooperative Content Delivery 179
6.4 EDCAM Algorithm 181
6.4.1 Energy-oriented Device System Profiling 181
6.4.2 Energy-efficient Quality Adaptation Algorithm 181
6.4.3 Two-party Cooperative Downloading 181
6.4.3.1 Motivation for Two-party Sharing 181
6.4.3.2 Content Sharing Mechanism 183
6.4.4 Overhead and Benefit Analysis 185
6.5 Simulation Testing and Result Analysis - EDCAM 189
6.5.1 EDCAM NS-3 Model and Simulation Testing Environment . 189
6.5.2 Simulation Scenario - Heterogeneous Wireless Network Environment: LTE and WiFi 190
6.5.2.1 Simulation Settings 191
6.5.2.2 Simulation Results 194
6.6 Chapter Summary 201

7 Conclusions and Future Work **203**
7.1 Overview 203
7.2 Contributions 204
7.3 Future Work 208

Bibliography **211**

List of Publications

- Ding, Ruiqi; Muntean, Gabriel-Miro, ”Device characteristics-based differentiated Energy-efficient Adaptive Solution for video delivery over heterogeneous wireless networks,” *Wireless Communications and Networking Conference (WCNC), 2013 IEEE* , pp.4588,4593, Shanghai, China, 7-10 April 2013

- Ding, Ruiqi; Muntean, Gabriel-Miro, ”A Novel device and application-aWare Energy efficient Routing Algorithm for WLANs,” *Globecom Workshops (GC Wkshps), 2012 IEEE* , pp.481,486, CA, USA, 3-7 Dec. 2012

- Ding, Ruiqi; Muntean, Gabriel-Miro, ”A context-aware cross-layer energy-efficient adaptive routing algorithm for WLAN communications,” *Local Computer Networks (LCN), 2012 IEEE 37th Conference on* , pp.176,179, FL, USA, 22-25 Oct. 2012

- Ding, Ruiqi; Muntean, Gabriel-Miro, ”An Energy-oriented Node Characteristics-Aware Routing Algorithm for wireless LAN,” *Broadband Multimedia Systems and Broadcasting (BMSB), 2011 IEEE International Symposium on* , pp.1,6, Germany 8-10 June 2011

Abstract

Energy efficiency is a key issue of highest importance to mobile wireless device users, as those devices are powered by batteries with limited power capacity. It is of very high interest to provide device differentiated user centric energy-efficient multimedia content delivery based on current application type, energy-oriented device features and user preferences. This thesis presents the following research contributions in the area of energy efficient multimedia delivery over heterogeneous wireless networks:

1. **ASP**: Energy-oriented Application-based System profiling for mobile devices: This profiling provides services to other contributions in this thesis. By monitoring the running applications and the corresponding power demand on the smart mobile device, a device energy model is obtained. The model is used in conjunction with applications' power signature to provide device energy constraints posed by running applications.

2. **AWERA**: A context-aWare cross-layer Energy-efficient adaptive Routing Algorithm for mobile devices in ad-hoc wireless LAN. AWERA takes its routing decisions based on current node energy constraint by making use of ASP. AWERA uses less energy constrained devices to relay data to other devices and therefore conserves the energy of either the more energy demanding devices or those with lower energy budgets.

3. **DEAS**: A Device characteristics-based differentiated Energy-efficient Adaptive Solution for video delivery over heterogeneous wireless networks. Based on the energy constraint, DEAS performs energy efficient content delivery adaptation for the current application. Unlike the existing solutions, DEAS takes all the applications running on the system into account and better balances QoS and energy efficiency.

4. **EDCAM**: Energy-efficient Device-differentiated Cooperative Adaptive Multimedia Delivery Solution. EDCAM is a hybrid innovative approach which combines multimedia quality adaptation and content sharing mechanisms to save energy at client devices. EDCAM relies on ASP for individual device energy constraints assessment.

5. A comprehensive survey on state-of-the-art energy-efficient network protocols and energy-saving network technologies.

Acknowledgements

First and foremost I want to thank my parents, especially my mother, who has always been the best mother and my inspiration since I do not even remember. She raised me with endless love, immense encouragement and condition-less support. The weekly phone conversion with her is a crucial element of my life in Ireland as it gives me strengthen every time I need it. She knows me better than myself and even the smallest step in my life is literally impossible without her. I would also like to thank all the relatives in my big family as I truly appreciate their love and support, which encouraged my adventure in Ireland.

I would like to thank my girlfriend Yang, who is the most attractive lady and more importantly, the best companion. This is not only because she is the only person who can appreciate my quirkiness and sense of humour, but she has always cheered me up on the darkest days and has gone through all my adventures in Ireland with me, including happiness, sadness, difficulties and even the smallest achievements. She has contributed so much to my academic achievement, my life and much much more. I believe she has made me a better person since our days in Wuhan University.

I want to thank my supervisor Gabriel Muntean. It has been an honour and pleasure to be his student. Call me biased, I think he is the most supportive and knowledgeable mentor one can have. He has taught me so much about research, thinking, writing, English, life and much more. I always struggle to organise my ideas rationally and present them clearly. Gabriel has patiently helped me to improve the way I think, write and present. His professionalism and love to work, his sincere kindness to everybody will be my lifetime inspiration. He literally gave me immense support whenever I need it no matter what I was facing. I am more than grateful for all the corrections he made to this thesis, every single paper, poster and presentation. He helped me to live up to my potential.

I gratefully acknowledge the funding sources that made my Ph.D. work possible. I was co funded by Irish Research Council and Citadel 100 Ltd. My life in Ireland was made possible by their generous support. Besides, Citadel 100 gave me a very interest subject to study. They always gave me plenty of time to do my job. I would like to express my biggest gratitude to Carl, Denis and George. Their hospitality and understanding are among the best things I have encountered in Ireland. I am truly grateful that they tolerated me to take my time to explore and learn. I gladly spread the words of theses good people and this great company in the conferences in different continents.

Many many thanks to the members of PEL at DCU. They have contributed immensely to my personal and professional time at DCU. Zhenhui is a dear friend. I am so grateful for his kind advices on personal and professional life in Ireland. As senior group members, Bogdan and Ramona helped me to get started with my PhD life. The China crows (Zhenhui, Shengyang, Ting and Longhao) and I went through the somewhat similar path from China to PEL at DCU so we always have a lot to talk and share. I learnt a lot about Ireland from Martin, Ronan and Lejla. Martin's expertise in Linux made my life easier as PEL's server admin. I enjoyed my chat and laugh with Irina, Ronan and Quang, it seems we are never running out of topics.

I want to thank the technical staffs of the school of Electronic Engineering, especially Robert Clare, a very kind, helpful and supportive man. I bothered him so many times for borrowing tools, printing posters, fixing devices and making 10 copies of door cards! I acknowledge the effort of Eduardo to build the prototype with me. He is my first student and it was a great pleasure working with him. Many thanks to Gabriel and Yang again for their effort to improve this thesis.

I gratefully acknowledge Prof. Zhili Sun, Dr. Xiaojun Wang, and Dr. Conor Brennan for making this happen.

I would like to thank my first car, the little Clio, which took me to every county of Ireland. Lastly, I want to thank Dublin City University and the great country of Ireland for all the resources and opportunities they gave me!

Dublin, November 2013

Ruiqi Ding

List of Tables

2.1 Energy efficient Network protocols. 27

2.2 Energy efficient Cross-layer Approaches. 42

2.3 Energy consumption at architectural level. 44

2.4 Graphical Display Energy Saving Techniques 51

2.5 Energy saving technologies at OS level. 55

2.6 Energy Efficiency at Application Layer. 70

3.1 Component Workload Profile Table 79

3.2 Application Profile Table 85

3.3 Error Rate of Models Using Different Target Value for Training 101

3.4 Error Rate of Models Using Different Training Applications 103

3.5 Error Rate of Playing ”Interview” At Different Quality Levels 105

3.6 Error Rate of Playing ”Surfing” At Different Quality Levels 105

3.7 Error Rate of Playing ”Top Gear” At Different Quality Levels 109

3.8 Error Rate of Utilisation-based Estimation and Power Signature-based Estimation 109

3.9 Overhead Results 110

4.1 Calculation of Individual Components’ Utility Functions 120

4.2 Real Readings in Idle State when GSM and WiFi Are Used 120

4.3 Simulation Settings for AWERA Scenario 1 130

4.4 AWERA Scenario 1: Results from Stationary Scenario 132

4.5 AWERA Scenario 1: Results from Mobile Scenario 133

4.6 Simulation Settings for AWERA Scenario 2 137
4.7 AWERA Scenario 2: Results from Simulation (Data Rate: 150 kbps) . 138
4.8 AWERA Scenario 2: Results from Simulation (Data Rate: 250 kbps) . 138
4.9 AWERA Scenario 2: Results from Simulation (Data Rate: 350 kbps) . 139
4.10 Simplified model used for testing 143
4.11 Simulation Settings for AWERA Scenario 3 144
4.12 AWERA Scenario 3: Results from Simulation (1Mbps) 146

5.1 Background Traffic Classes for DEAS Scenario 1 158
5.2 Simulation Settings for DEAS Scenario 1 160
5.3 DEAS Scenario 1: Simulation Results for NonAd 162
5.4 DEAS Scenario 1: Simulation Results for SAMMy 163
5.5 DEAS Scenario 1: Simulation Results for DEAS 163
5.6 Background Traffic Classes: Number of Background Traffic Subscribers for DEAS Scenario 2 164
5.7 Simulation Settings for DEAS Scenario 2 166
5.8 DEAS Scenario 2: Simulation Results for DEAS 168
5.9 DEAS Scenario 2: Simulation Results for ESTREL 168
5.10 DEAS Scenario 2: Simulation Results for NonAd 168

6.1 Parameter Definitions 186
6.2 Background Traffic Classes: Number of Background Traffic Subscribers for EDCAM Scenario 192
6.3 Simulation Settings for EDCAM 193
6.4 EDCAM Scenario: Simulation Results for EDCAM 195
6.5 EDCAM Scenario: Simulation Results for DEAS 195
6.6 EDCAM Scenario: Simulation Results for ESTREL 196
6.7 EDCAM Scenario: Simulation Results for NonAd 196

List of Figures

1.1 Mobile users in a heterogeneous wireless network environment 2

1.2 Relation between Four Contributions 6

2.1 Layered Protocols Model 13

2.2 Classification of Energy Efficient Network Protocols 15

2.3 Minimum Battery Cost Routing 17

2.4 Low Energy Adaptive Clustering Hierarchy (LEACH). 20

2.5 Geographic Adaptive Fidelity Protocol 21

2.6 Illustration of General Cross-layer Approach. 31

2.7 Illustration of Traffic Shaping. 36

2.8 Illustration of Application-specific Network Management for Energy-aware Streaming. 36

2.9 PEDAMACS: Joint Routing and MAC Layer Sleep Scheduling. 38

2.10 Classification of Application Level Optimization Techniques. 60

2.11 Illustration of Adaptive Control at Application Level. 63

2.12 Illustration of Partial Caching. 67

3.1 The Architecture of ASP 77

3.2 The Architecture of Neural Network for Regression 84

3.3 Graphical User Interface 88

3.4 Class Diagram of Java Implementation 89

3.5 Real Tests : Set-up of Devices 91

3.6 Relation between Battery Discharge and Screen Brightness 92

3.7 Relation between Battery Discharge and CPU Workload 93
3.8 Relation between Battery Discharge and CPU Frequency 94
3.9 Relation between Battery Discharge and CPU Workload - Two-stage Frequency Setting 94
3.10 Relation between Battery Discharge and CPU Workload without Knowing Frequency Settings 95
3.11 Relation between Battery Discharge and CPU Frequency Settings . . 95
3.12 Relation between Battery Discharge and Bit Rate (Outbound Traffic) . 97
3.13 Relation between Battery Discharge and Packet Rate (Outbound Traffic) 97
3.14 Relation between Battery Discharge and Bit Rate (Inbound Traffic) . . 98
3.15 Estimations of Models Using Power as Training Target 102
3.16 Estimations of Models Using Current as Training Target 102
3.17 Estimations of Models Trained Using Predefined Tasks 104
3.18 Estimations of Models Trained Using Daily Used Applications 104
3.19 Estimations of Playing High Quality ”Interview” 105
3.20 Estimations of Playing Low Quality ”Interview” 106
3.21 Estimations of Playing High Quality ”Surfing” 106
3.22 Estimations of Playing Low Quality ”Surfing” 107
3.23 Estimations of Playing High Quality ”Top Gear” 107
3.24 Estimations of Playing Medium Quality ”Top Gear” 108
3.25 Estimations of Playing Low Quality ”Top Gear” 108
3.26 Utilisation-based Estimations (SVR) and Power Signature-based Estimations 110

4.1 Example of Use Case Scenario for AWERA - Theme Park 115
4.2 The Architecture of ASP and AWERA 117
4.3 Cross-layer Mechanism of AWERA Working with ASP 119
4.4 Class Diagram of AWERA Implementation in NS-2 126
4.5 Simulation Topology - AWERA Scenario 1 128

4.6 Average Remaining Energy for Nodes Performing Multimedia Delivery and Web Browsing, respectively - In Stationary Case (J) 130
4.7 Average Throughput - In Stationary Case (Kbps) 131
4.8 Average Delay - In Stationary Case (Sec) 131
4.9 Average Remaining Energy for Nodes Performing Multimedia Delivery and Web Browsing, respectively - In Mobile Case (J) 131
4.10 Average Throughput - In Mobile Case (Kbps) 132
4.11 Average Delay - In Mobile Case (Sec) 132
4.12 Simulation Topology - AWERA Scenario 2 135
4.13 AWERA Scenario 2: Average Energy Consumption of Nodes from Class A (J) 139
4.14 AWERA Scenario 2: Average Energy Consumption of Nodes from Class B (J) 139
4.15 AWERA Scenario 2: Average Energy Consumption of Nodes from Class C (J) 140
4.16 AWERA Scenario 2: Average End-to-end Delay (Sec) 140
4.17 AWERA Scenario 2: Average Throughput of Receiver Nodes (Kbps) . 140
4.18 Simulation Topology - AWERA Scenario 3 142
4.19 AWERA Scenario 3: Average Energy Consumption of Class A Nodes (J) 144
4.20 AWERA Scenario 3: Average Energy Consumption of Class B Nodes (J) 145
4.21 AWERA Scenario 3: Average Energy Consumption of Class C Nodes (J) 145
4.22 AWERA Scenario 3: Average End-to-end Delay (Sec) 145
4.23 AWERA Scenario 3: Average Throughput of Receiver Nodes (Kbps) . 146

5.1 Use Case Scenario of DEAS 150
5.2 Block-level Architecture of DEAS 151

5.3 Class Diagram of DEAS Implementation in NS-3 157
5.4 Simulation Topology - DEAS Scenario 1 159
5.5 DEAS Scenario 1: Packet Loss Rate (%) 161
5.6 DEAS Scenario 1: Energy Consumption (J) 161
5.7 DEAS Scenario 1: Quality - PSNR (dB) 162
5.8 Simulation Topology - DEAS Scenario 2 164
5.9 DEAS Scenario 2: Energy Consumption (J) 166
5.10 DEAS Scenario 2: Quality - PSNR (dB) 167
5.11 DEAS Scenario 2: Jitter (Sec) 167
5.12 DEAS Scenario 2: Delay (Sec) 168

6.1 The Latest Heterogeneous Wireless Networks 172
6.2 Use Case Scenario of EDCAM 176
6.3 Different Multimedia Content Delivery Strategy 177
6.4 ASP Mechanism 177
6.5 Architecture of EDCAM 179
6.6 Sharing Partner Selection 180
6.7 Handshake Procedure to Establish a Connection with the Partner for Content Sharing 185
6.8 Class Diagram of EDCAM Implementation in NS-3 189
6.9 Simulation Topology - EDCAM 190
6.10 EDCAM Scenario: Remaining Energy - 1Mbps Background Traffic . 197
6.11 EDCAM Scenario: Remaining Energy - 2Mbps Background Traffic . 197
6.12 EDCAM Scenario: Average Delay - 1Mbps Background Traffic . . . 198
6.13 EDCAM Scenario: Average Delay - 2Mbps Background Traffic . . . 198
6.14 EDCAM Scenario: Average Jitter - 1Mbps Background Traffic 199
6.15 EDCAM Scenario: Average Jitter - 2Mbps Background Traffic 199
6.16 EDCAM Scenario: Average PSNR - 1Mbps Background Traffic . . . 200
6.17 EDCAM Scenario: Average PSNR - 2Mbps Background Traffic . . . 200

Abbreviations and Acronyms

3G	Third Generation of Mobile Telecommunications Technology
4G	Fourth Generation of Mobile Telecommunications Technology
ACK	Acknowledgement
AID	Association ID
AMPS	Advanced Mobile Phone System
AP	Access Point
ARQ	Automatic Repeat Request
ASN	Access Service Network
AUC	Authentication Centre
BE	Best Effort
BSC	Base Station Controller
BSS	Base Station Subsystem
BSS	Base Service Set
BTS	Base Transceiver Stations
CBR	Constant Bit Rate
CDMA	Code Division Multiple Access
CDN	Content Delivery Network
CN	Core Network
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Clear-To-Send

DCCP Datagram Congestion Control Protocol

DCF Distributed Coordination Function

DCT Discreet Cosine Transform

DIFS DCF Interframe Space

DSSS Direct-Sequence Spread Spectrum

EDGE Enhanced Data rates for Global Evolution

EIR Equipment Identification Register

ertPS extended real-time Polling Service

ETSI European Telecommunications Standards Institute

FDMA Frequency Division Multiple Access

FEC Forward Error Correction

FR Full Reference

FTP File Transfer Protocol

GPRS General Packet Radio Service

GSM Global System for Mobile Communications

HLR Home Location Register

HTTP Hypertext Transfer Protocol

IBSS Independent BSS

ICMP Internet Control Message Protocol

ICT Information and Communications Technologies

IEEE Institute of Electrical and Electronics Engineers

IMAP Internet Message Access Protocol

IMT-2000 International Mobile Telecommunications-2000

IP Internet Protocol

IPv4 Internet Protocol version 4

IPv6 Internet Protocol version 6

ITU International Telecommunications Union

JPEG Joint Photographic Experts Group

MAC Medium Access Control

MIMO Multiple Input Multiple Output

MOS Mean Opinion Score

MPEG Moving Pictures Experts Group

MSC Mobile Service Switching Centre

MSS Maximum Segment Size

NAV Network Allocation Vector

NMT Nordic Mobile Telephony

NR No Reference

nrtPS non-real-time Polling Service

NSP Network Service Provider

NSS Network Switching Subsystem

OFDM Orthogonal Frequency-Division Multiplexing

OMC Operation Maintenance Centre

OSS	Operation and Support System
PAT	Packet/ApplicaTion manager
PCF	Point Coordination Function
PDA	Personal Digital Assistants
PDV	Packet Delay Variation
PESQ	Perceptual Evaluation of Speech Quality
PEVQ	Perceptual Evaluation of Video Quality
POP	Post Office Protocol
PSM	Power Saving Mode
PSQM	Perceptual Speech Quality Measurement
QoE	Quality of Experience
QoS	Quality of Service
RR	Reduced Reference
RTCP	RTP Control Protocol
RTP	Real-Time Transport Protocol
RTT	Round Trip Time
rtPS	real-time Polling Service
RTS	Request-To-Send
RTSP	Real Time Streaming Protocol
Rx	Receive
SCTP	Stream Control Transmission Protocol

SDP Session Description Protocol

SIFS Short Inter-Frame Space

SIM Subscriber Identity Module

SMTP Simple Mail Transfer Protocol

STELA Slow sTart Exponential and Linear Algorithm

TCP Transmission Control Protocol

TDMA Frequency Division Multiple Access

TIM Traffic Indication Map

Tx Transmit

UDP User Datagram Protocol

UGS Unsolicited Grant Service

UMTS Universal Mobile Telecommunications System

UE User Equipment

UTRAN UMTS Radio Access Network

VBR Variable Bit Rate

VLR Visitor Location Register

VoIP Voice over IP

WCDMA Wideband Code Division Multiple Access

WiFi Wireless Fidelity

WiMAX Worldwide Interoperability for Microwave Access

WLAN Wireless Local Area Networks

WMAN Wireless Metropolitan Area Network

WNIC Wireless Network Interface Card

WPAN Wireless Personal Area Network

WWAN Wireless Wide Area Network

Chapter 1

Introduction

1.1 Research Motivation

In recent years, the applications relying on wireless communications have extended their usage from narrow industrial areas to people's daily life. The latest innovative digital mobile devices, such as smart phones and tablet PCs, are bringing the extensive developments of wireless communication technologies in the consumer device sector. This has exponentially increased the number of people connecting to the Internet through cellular and broadband wireless networks via mobile devices. People can work on such devices connected wirelessly to the Internet from homes, restaurants or even travelling trains. Some international conferences are held over the Internet to save costs and reduce CO2 emissions.

According to statistical results from industry reports and surveys, of the world's 4 billion mobile phones in use, over 1.08 billions are smart phones, and half of the local searches are performed on mobile devices [1].

Smart devices are equipped with WiFi interfaces to connect to the wireless LAN infrastructure, often deployed in family homes, shopping malls or theme parks [2] in order to provide mobile users' access to information and complex applications. This

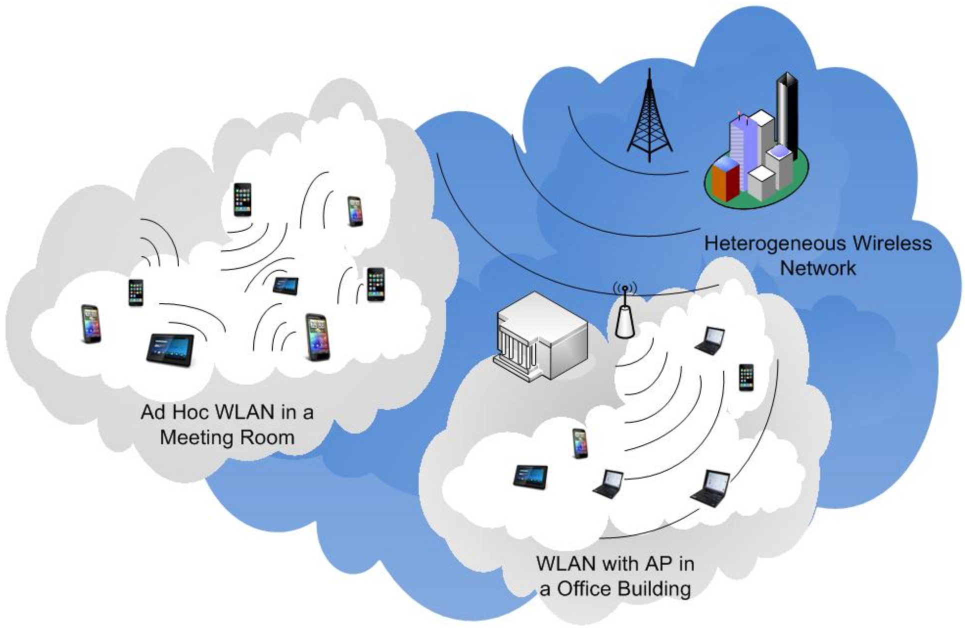

Figure 1.1: Mobile users in a heterogeneous wireless network environment

kind of wireless LAN may work alone or act as part of a heterogeneous network environment connected to the Internet. Many mobile devices have also cellular network interfaces to enable their connectivity to 3G/3.5G/4G networks in areas where broadband is not deployed. Cellular networks and wireless MAN/LANs are often used together to provide anywhere and any time connectivity. Therefore the coexistence of various network types as part of heterogeneous network environments is envisaged as the future in wireless networking. A hybrid ad-hoc and infrastructure network environment is illustrated in Fig.1.1.

Along with the increasing coverage of the wireless networks, the emerging rich media features of the latest services also contribute to the popularity of the smart devices. The latest digital mobile devices support complex rich media applications which offer excellent user experience and large range of services, including file sharing, on line banking, family entertainment, online social networks, GPS navigation, and much more. Notably, recently more than half of the Internet searches were performed from mobile devices [1] and multimedia-based traffic accounts for 49 percent and 53 percent of the total data consumption over smartphones and tablets, respectively [3]. This trend shows how people are increasingly using mobile wireless-enabled services.

The massive development and deployment of wireless networks have resulted in new demands and new challenges with regard to energy efficiency and performance. Energy efficiency is investigated both for the content providers [4] and client devices. On one hand, people demand more powerful devices and faster networks to accommodate multimedia content delivery and complex applications, and these demands require more energy expenses. One the other hand, people expect devices to have slim shapes and large capacity battery to support the device running longer periods between recharges without compromising mobility. This situation imposes challenges to the design of the energy efficient multimedia content delivery with regard to both the hardware (e.g. network infrastructure, smart phones) and software (e.g. applications, networking protocols). Moreover, increasing amounts of multimedia content are delivered and will be exchanged over the future wireless networks, which requires new protocols and new solutions in order to suit the high bit rate, high timing constraint network traffic delivery.

As indicated by Martin et al [5], there is a need for a more sophisticated approach that takes into account the nature of the current application and current battery levels, analyses and processes the data and adjusts the delivery process in an innovative manner in order to prolong the battery life in real time for mobile devices. In this thesis, we address this issue and present a suite of energy efficiency solutions which include an energy efficient routing algorithm for mobile devices over ad-hoc and a cooperative adaptive multimedia delivery solution for mobile devices. Their Device-Differentiation feature is made possible by a novel Energy-oriented System Profiling process, a self-constructive application-aware energy modelling technique for mobile devices. The algorithms have been tested with both simulations and a prototype system that has been specifically developed.

1.2 Problem Statement

With the increasing deployment of heterogeneous wireless networks and fast development of smart devices, energy efficient multimedia content delivery becomes an important element in wireless networking. This observation is backed by concrete data: Cisco estimates handsets will generate in excess of 50 percent of mobile data traffic in 2014, and the video content will account for more than two thirds of the globe networking traffic by 2016 [6]. This trend is made possible by the multimedia capability support and ubiquitous network connectivity provided by the latest smart mobile devices.

Energy efficiency has always been a key issue in wireless communication. First, wireless communication requires cable free devices, which offer constant connectivity even on the move. In this case, the mobile devices are powered by battery with limited capacity rather than mains power supply. Secondly, smart phone and tablet PC manufactures upgrade the devices with the more energy demanding faster processors and larger screens every year, while the revolution of the battery capacity does not take place at the same pace. To increase device lifetime by installing larger batteries would make people reluctant to carry and use mobile devices so often. In conclusion, energy constraint is always a big challenge for wireless networks and wireless devices.

In addition, as Information and Communications Technologies (ICT) sector is still developing all over the world, energy conservation has become a global environmental issue. Technology analysts claims that the ICT industry contributes with 2 percent to the global carbon emissions, which is approximately equivalent with that of the aviation industry. Moreover, it is estimated that ICT will be responsible for 3 percent of global emissions by 2020. However, the use of ICT can also increase energy efficiency and reduce emissions in other sectors, for example, transportation [7]. Therefore smart use of novel ICT technologies can determine reductions in CO2 emissions as well.

To achieve energy efficiency, people optimise device functionalities, application features and wireless networking protocols. For example, the CPU of smart phones will shut down part of its components when the work load is light; some smart phone applications sense the lighting conditions in the environment and adjust the back light brightness of the screen accordingly; some energy efficient routing and application layer protocols are proposed for energy saving during wireless data transmissions.

As smart devices gain more and more popularity all over the world, the content of networking traffic is changing during the years. For example, more multimedia content traffic is being delivered. In the old days of the Internet, the tiny bandwidth and slow computing capability determined the content of network traffic to be text dominated, for example, made of emails and Unix commands. Nowadays, VoIP, on line video chat, on line video sharing, and interactive gaming push the limit of the device as well as the wireless networks by exchanging rich media content. This multimedia content forms a very important element in the context of mobile device usage. In this sense, successful and efficient multimedia content transmission plays an important role in today's networks.

In this context, there is a need to keep record of the device energy constraint information. This information is utilised by carefully designed protocols to provide device differentiated service for energy efficiency benefit of both the device and network. Maintaining energy constraint information and designing such protocols are problems. In this thesis, various device-differentiated energy-efficient solutions are presented and discussed. The major contributions are outlined in the next section.

1.3 Solution Overview and Contributions

This thesis presents ASP, an intelligent automatic "energy-oriented Application-based System Profiling" process that builds an incrementally updated energy model for individual mobile devices. The process calculates the energy constraints of smart motile devices on demand. Using the energy constraint information provided by ASP, a

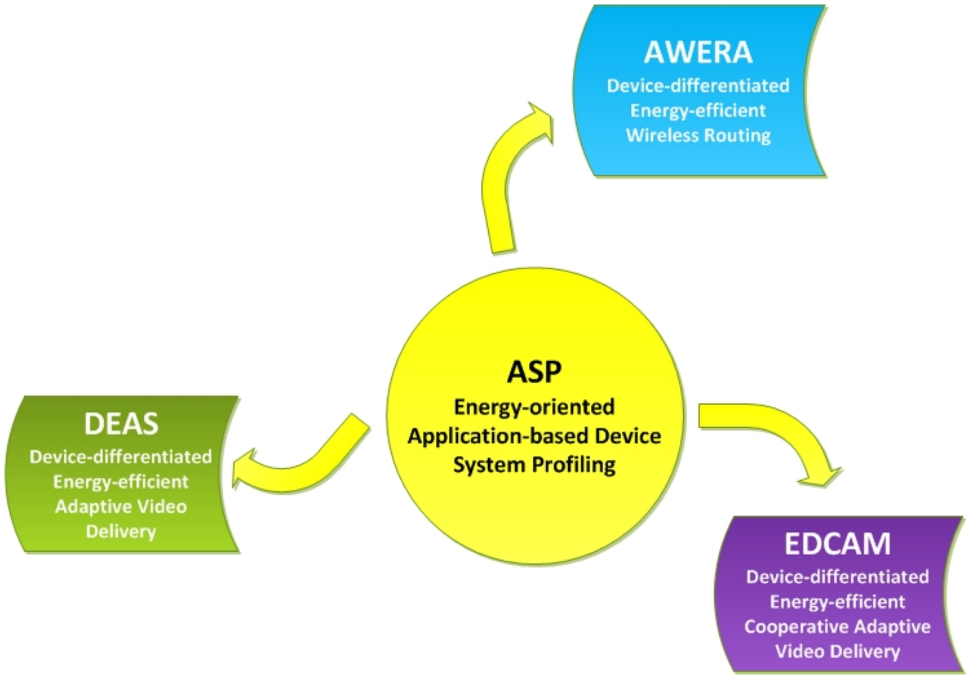

Figure 1.2: Relation between Four Contributions

Context-aware Differentiated Cross-layer Energy-efficient Routing solution (AWERA), and a Device-differentiated Adaptive Video Delivery solution (DEAS) were proposed and presented in this thesis. Based on DEAS, an Adaptive Energy Efficient Cooperative Content Delivery solution (EDCAM) was proposed to utilise the multi-home capability of model mobile devices for further energy savings. The relation between these four contributions is depicted in Fig.1.2.

Various applications, devices and wireless links yield different contexts, and these contexts affect both performance and energy efficiency. AWERA keeps record of device energy constraint using energy model and application power signature to differentiate devices for energy efficient ad-hoc routing which conserves more energy for relatively energy vulnerable devices.

DEAS monitors the active video application in terms of perceived quality, as estimated using PSNR, and calculates device energy constraint using ASP. The information is then used by an energy-efficient quality adaptation algorithm to request the server to perform quality adjustments to the content delivery. This solution conserves energy by requesting the less energy demanding lower quality content for devices with tight energy constraints.

EDCAM is a hybrid innovative content delivery solution that combines multimedia quality adaptation and content sharing mechanisms to save energy at client devices. EDCAM relies on ASP to assess individual *device energy constraint*. These constraints along with QoS delivery scores are used as metrics for the multimedia delivery quality adaptation. Each device will then search for a content sharing partner from which to retrieve the content, and therefore reduces the usage of high energy consuming networks and increases energy efficiency.

1.3.1 Energy-oriented Application-based System Profiling - ASP

According to the latest research [8] [9], the device energy constraints vary significantly for different applications. Besides, network traffic processing is one of the most energy consuming tasks for the mobile devices. Based on these results, ASP's goal is to record the power signatures in terms of typical power of mobile device for different applications and maintains such information in the form of an application-based system profile. The profile is updated along the lifetime of the mobile device usage. Additionally, a regression-based automatic energy modelling technique is used to generate the energy model for individual devices. Knowing the running applications' type, the energy model along with the application profile will generate the current device energy constraint on demand.

This effort creates a learning mechanism for the device to adapt to the context environment automatically and reduces the overhead of monitoring. An initialisation phase is conducted to each device once only in order to make the learning process device independent. The contribution is novel as ASP improves existing utilisation-based models by using application signatures for inexpensive and easy energy constraint estimations. For proof of concept purposes, ASP was implemented as a prototype system on an Android platform.

1.3.2 Context-aware Differentiated Adaptive Routing - AWERA

Based on ASP, AWERA obtains and maintains context information, including link quality of the available links and energy constraint of the available devices. The context information makes possible to differentiate devices according to their context of usage, in order to achieve prioritised routing/energy saving in ad-hoc routing scenarios. Moreover, the periodical updating mechanism enables adapt the routing strategy to any change in the context. In a mobile device formed ad-hoc wireless LAN, AWERA encourages devices with high energy budgets to act as intermediate nodes for data relaying in order to save energy at the devices with tight energy budgets.

Based on the energy model provided by ASP and context information, an adaptive routing protocol is specially designed for energy efficient wireless routing. First, the context information is passed on to the network layer. In addition, link quality of the wireless links to the available neighbours is also considered in the routing process. Consequently, a dedicated routing information processing component converts the context information and link layer feedback into a new metric, which is maintained in a special routing table to accommodate such energy constraint-related context information. The routing table will be updated periodically to suit any changes in the context. Such cross-layer effort makes AWERA effective in adapting to the change of context and efficient in energy conservation. Consequently, AWERA makes routing decisions intelligently and adaptively so that the most energy efficient routes are used during wireless data transmissions.

1.3.3 Device-Differentiated Adaptive Content Delivery - DEAS

The limited battery capacity of current mobile devices and increasing amount of rich media content delivered over wireless networks have driven the latest research on energy efficient content delivery over wireless networks. Many energy-aware research solutions have been proposed, such as traffic shaping, content adaptation, content sharing, etc. The existing solutions focus on the delivery application without considering application running environment and device features that put different energy constraints on the whole content delivery process. DEAS based on ASP's profile, performs energy efficient content delivery adaptation for the current multimedia delivery application. When the current energy budget is low, for example, when the device is running other energy consuming applications or the battery energy level is low, DEAS will request the content provider to transmit lower quality content to save energy. This is because lower quality content requires less energy for decoding and data transmission. While the energy budget becomes higher, for example, when the other energy consuming application is closed, DEAS will request the content provider to transmit the multimedia content of higher quality.

1.3.4 Energy-efficient Device-differentiated Cooperative Adaptive Multimedia Delivery - EDCAM

Energy-efficient multimedia content delivery to mobile devices has always been an important subject. Since the increasingly affordable and powerful smart mobile devices are equipped with multiple networking interfaces, the multi-home networking capability of mobile devices has to be addressed to improve device energy efficiency. However, content sharing technique utilising multiple networking interfaces is mostly investigated for server bandwidth saving instead of client energy saving. Meanwhile content provider and client propose potential for device-based energy saving by offering the same content at different quality levels and by quality adaptation (see DEAS).

In this context, EDCAM is proposed as a hybrid solution which combines multimedia content quality adaptation with content sharing aiming at saving energy at client devices. It uses ASP for energy-oriented application-based system profiling. The energy constraint from ASP along with QoS scores are used as metrics for quality adaptation while the device is receiving multimedia content. The device will then search for a content sharing partner to download the content collaboratively for reducing cellular interface usage and better energy efficiency.

EDCAM's sharing-assisted adaptive delivery introduces two-party content sharing to address the above problems. The device establishes a simple ad-hoc WiFi connection to a partner downloading the same content of the same quality. Secondly either device downloads half of the content and fetches the other half from each other. Therefore, the group management overhead is minimised.

1.4 Thesis Structure

This thesis is organised as such:

- Chapter 1 introduces the research motivation and the major contributions.
- Chapter 2 presents background knowledge and related works on energy efficient solutions at network layer, application layer, mobile platforms as well as cross layer energy saving solutions.
- Chapter 3 introduces ASP, the energy-oriented application-aware system profilling technique used to provide device energy constraint information to other proposed solutions. Then it presents the prototype implementation of ASP and the real test results, including scenario set-up, testing strategy, energy model validation and results analysis.
- Chapter 4 introduces AWERA and explains its architecture and algorithm, including routing cost calculation and routing mechanism. Then it presents the simulation test bed description and testing results for AWERA in comparison with MBCR and AODV.
- Chapter 5 introduces DEAS and explains its architecture and algorithm includ-

ing battery life time calculation and quality adaptation mechanism. Then it presents the simulation test bed description, and testing results for DEAS in comparison with SAMMy, ESTREL and No adaptation.

- Chapter 6 introduces EDCAM and explains its architecture and algorithm including sharing partner selection and content sharing mechanism. Then it presents the simulation test bed description, and testing results for EDCAM in comparison with DEAS, ESTREL and No adaptation.

- Chapter 7 draws conclusions and shows the future work.

Chapter 2

Background and Related Works

2.1 Layered Protocols Model

Network technology makes end-to-end communication possible by delivering data from the source device to the destination device. Sometimes data has to travel along different networks and devices to reach its final destination. In order to meet performance requirements and maintain connectivity and compatibility, researchers designed various protocols to regulate the behaviour of network devices while sending, receiving and forwarding data. These protocols are distributed over a protocol stack consisting of various layers, each layer offering services to the layer above. In this chapter, we focus on a five-layer-model where the protocol stack from top down, is divided into application layer, transport layer, network layer, data link layer including Medium Access Control (MAC) layer and Physical layer as illustrated in Fig. 2.1.

Application layer implements various protocols used by user applications such as video streaming, file sharing (FTP), web surfing (HTTP), etc. It governs end-to-end application-specific data communication transparently. From the viewpoint of application layer, applications are communicating with each other directly. Application's running status can be adjusted to conserve energy and collect user response to support

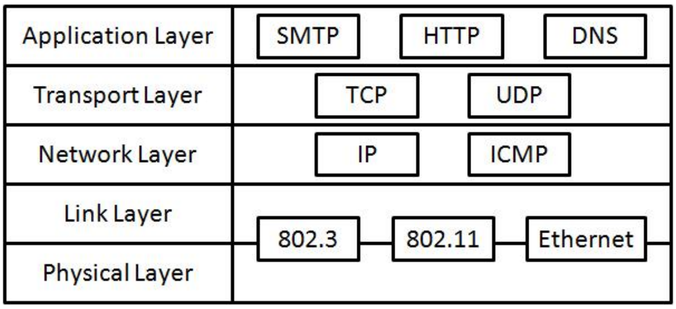

Figure 2.1: Layered Protocols Model

QoS.

Transport layer ensures end-to-end data transfer between source-host and destination-host by establishing and maintaining the connection between the pair of hosts. It provides a simplified networking interface to each host without the complexity of network topology. Retransmission mechanisms and congestion control can be tuned to support QoS.

Network layer maintains an identifier for each host in the network and performs routing by directing packets to the right network device, namely the router/host closer to the destination, so that packets could cross one or more networks to reach the destination. IP hand-off can be performed for mobile hosts to maintain connectivity. Routing protocols can apply energy-aware and performance-related metrics while making routing decisions.

The protocols at data link and especially at MAC layer, a sub layer of link layer, are deployed at a lower layer and govern the access to communication medium ensuring packet exchange between direct linked devices. Improved MAC layer protocol may effectively improve energy efficiency and performance.

Physical layer takes care of the very low level bit-by-bit data transmission.

Energy saving techniques and QoS support features can be implemented into protocols at each individual layer, including cross layer approaches where protocols at different layers can easily interact with each other exchanging information regarding link quality, energy consumption, error rate and various other types of parameters.

2.2 Energy-Efficient Network Layer

Network layer maintains the identifier of each host within the network (i.e. IP address) according to which routing protocols identify each individual host in order to be able to find a path from a source to the destination over one or more networks. It can effectively reduce the energy consumption of the networks by reducing the energy consumption at relaying devices and router. For example, **EABF** [10] and [11] effectively improve the energy efficiency of packet forwarding devices from the view point of hardware design. Energy efficient routing protocols achieve the same goal of saving energy from the view point of routing protocol design. Routing protocols gather information such as distance between hosts, link quality, bandwidth availability and number of intermediate hosts. Based on this information, possible routes interconnecting each pair of hosts are calculated. A desirable route can then be selected if any available. This can be done in a distributed fashion by allowing hosts to make their own decisions, or have a centralized decision making approach.

Transmitting a packet from a source host to the destination host is similar to navigating through a big city or event between locations in distinct cities. For a big city, it is quite common to have a number of buses that could take the passenger to the destination through different routes. Among those routes, there will be a route taking least time, a route having shortest distance, a route having least gas consumption, and so on. Normally, people would go for the one taking least time, namely go for the routing protocol that can deliver data as soon as possible with shortest delay. However, to find the most energy efficient route could make more sense in certain scenarios due to the fact that each host may have different energy constraints (remaining energy level, battery capacity, energy depletion rate, etc).

In order to achieve better energy efficiency for the whole network, energy aware

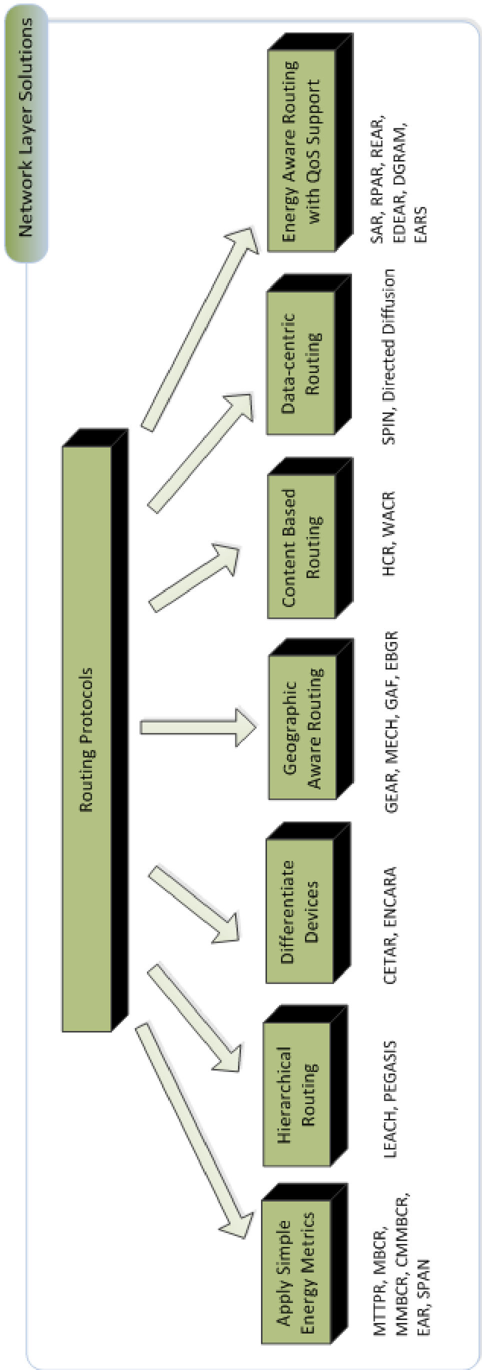

Figure 2.2: Classification of Energy Efficient Network Protocols .

metrics can be used by the routing process as well as geographic information. The hierarchical structure of mobile hosts also promotes better performance and energy efficiency. In addition, more recent routing techniques including data-centric routing and content-based routing are content centric and application-aware, which is a handy feature for performance optimization. Besides, QoS support can be embedded into the routing process by traffic differentiation and load balance at network level.

The classification of energy efficient network layer protocols is depicted in Fig. 2.2. The following sections present various energy-aware packet routing techniques.

2.2.1 Energy-aware Routing

Energy-aware routing is of paramount importance especially for applications involving ad-hoc multi-hop wireless networks where the limited energy capacity of most typical wireless devices brings this challenge forward. The amount of energy consumed to transmit a packet between two hosts depends on the link quality and state. In the context of ad-hoc networks where two distant hosts can be connected over distinct links and routes an energy efficient routing protocol can conserve energy by specifying a certain route from the source to the destination based on energy considerations.

Minimum Total Transmission Power Routing (MTTPR) [12] takes the most straight forward approach by selecting the route with the least total energy cost. It usually takes several hops from the source to the destination, and each hop requires a certain amount of energy to transmit a packet. For every candidate path from the source host to the destination, MTTPR collects and adds up the energy cost of each hop to give a total cost for each path, and consequently selects the path with the least total energy consumption.

Besides the amount of consumed energy, the capacity and remaining energy level of battery could also be a metric for energy aware routing protocols, because devices with more remaining battery could survive longer and forward more incoming packets than those with little remaining battery capacity. Instead of total energy consumption, Minimum Battery Cost Routing [13] chooses the total remaining battery capacity of every device along each path as the metric and gives the total remaining battery capac-

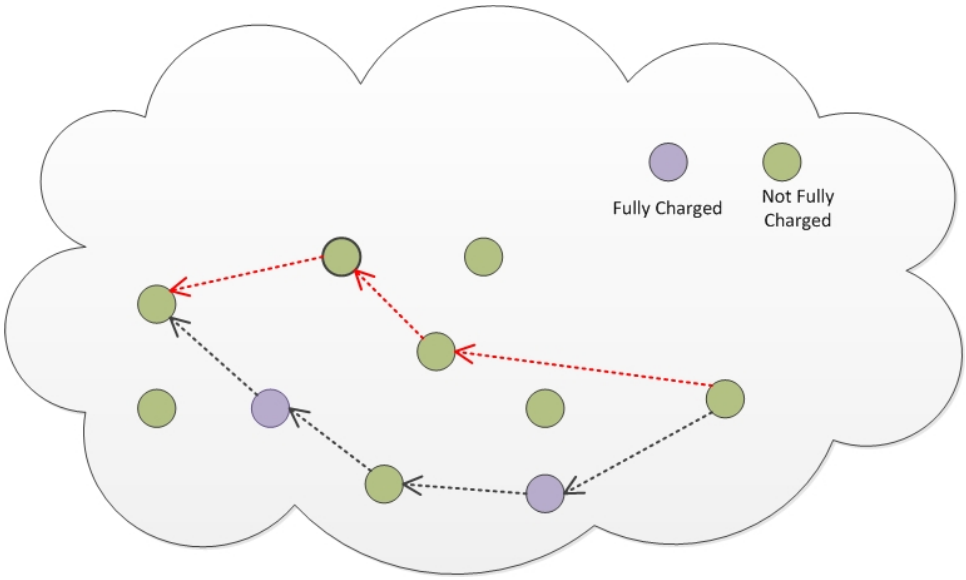

Figure 2.3: Minimum Battery Cost Routing

ity of each route. It compares the battery capacity of all possible routes and uses the one with devices along it of most remaining energy. This is shown in Fig.2.3, where the black path containing more hops might be selected as an optimal on other than the red path, due to the fact that the black path includes two fully charged nodes, which makes the black path have more total remaining energy than the red one. Because MBCR considers a path as one entity, it fails to anticipate such fact as a path with most battery remaining could include a device with very little remaining energy. If a route as such is used, it will drain the energy of the ”weak device” and becomes broken path soon.

To address this issue **Min-Max Battery Cost Routing (MMBCR)** [13] checks the remaining battery level of each device to avoid involving devices with low power along the path.

Conditional Min-Max Battery Cost routing [13] is a hybrid solution mixing MTTPR and MMBCR. The remaining energy of each device along the path is considered, together with the total energy consumption for each path. The two factors combined determine the cost of each route.

Arezoomand et al. [14] improved MBCR by recording the hop count of each route as well as the total energy consumption, thus avoiding devices with little energy since these will bring down the average remaining energy per hop.

Energy Aware Routing (EAR) [15] is a more sophisticated solution which considers both energy cost of each hop and the residual energy level of each host. The normalized values of these two measures are used by EAR to compute and select routes. Based on these values a host chooses the neighbor with the highest energy level and lowest link energy cost. Consequently, EAR achieves efficient energy distribution among hosts and avoids unbalanced energy consumption.

Span [16] is a distributed algorithm which allows each host to make its own decisions regarding sleep schedule. By evaluating the available energy and the contribution it can make to the overall network connectivity (the more pair of hosts it can connect, the more it can contribute) it decide if it should stay in active mode or switch to sleep mode. Therefore Span is a localized routing solution that decreases routing overhead, scales well and has short route computing delay.

Wei et al. [17] presents an energy saving dynamic relaying scheme in cooperative networks using finite-state Markov decision process based prediction method. The Markov decision process is used to predict the upcoming channel state, relay selection and transmission power selection. The formulated decision process results in energy efficient relay selection that makes more efficient use of wireless channels.

2.2.2 Hierarchical Routing

For infrastructure-based wireless networks, hosts need to communicate with base stations regardless of the distance. However, routing protocols could organize hosts into clusters and elect a head for each cluster so that only that particular host has to communicate directly with the base station. In this manner, the rest of the hosts within the same cluster only communicate with the head host, which is closer and accessing it is more energy efficient.

Low Energy Adaptive Clustering Hierarchy (LEACH) [18, 19] is a cluster-based solution as depicted in Fig. 2.4.

PEGASIS [20, 21] is a chain-based hierarchical routing algorithm. Each non-head host only exchange packets with its two direct neighbors forming a chain for each cluster. Compared with LEACH, this mechanism further reduces energy consumption since it requires even less energy to communicate with direct neighbors instead of the cluster head.

The main disadvantages of the above solutions include the head host depleting battery faster due to extra workload to aggregate packets from every cluster member and communicate with the base station on their behalf. On the other hand, single tier clustering may not be as effective as expected for wide area routing.

To address these issues, Liu et al. [22] presents a multi-levels cluster-based routing solution for wireless vehicular sensor networks which rotates the holder of the cluster head duty and offers multi-tier cluster hierarchy. It outperforms LEACH in terms of stability and energy efficiency for long range communications. Furthermore, multi-tier clustering makes large scale deployment possible, which is suitable for vehicular network applications.

2.2.3 Differentiate Devices

Energy Type Aware Routing (CETAR) [23] differentiates devices based on their role in data exchange: receiver or sender. A device sending data packets to other hosts for a long period is considered an active sender. This device consumes more energy with data generation and forwarding. CETAR discourages active sender to act as routers in order to conserve the energy of these active senders.

In heterogeneous wireless environments, devices of different types may join the

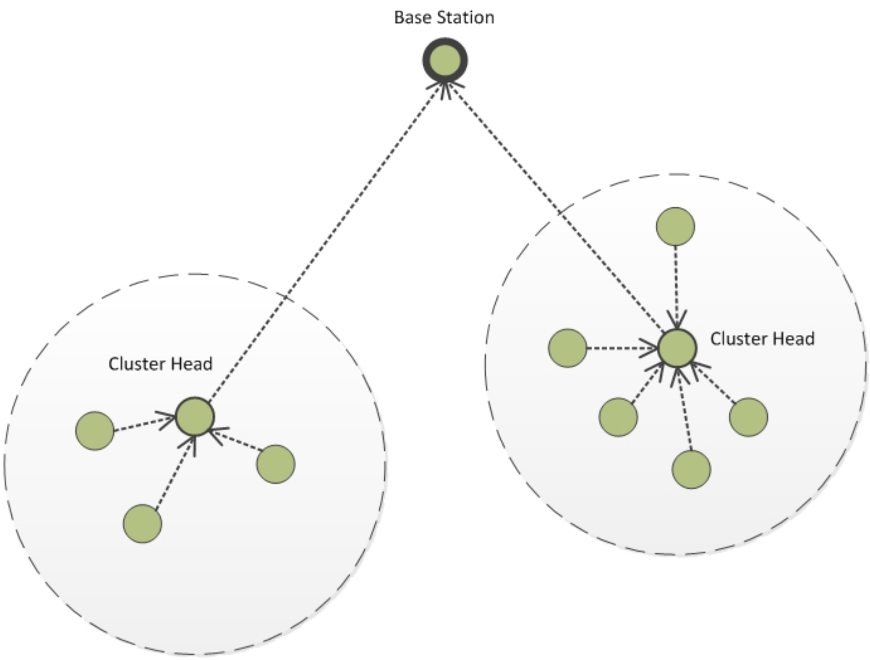

Figure 2.4: Low Energy Adaptive Clustering Hierarchy (LEACH).

same wireless network regardless of their software environment, power source, battery quality, etc. For instance, a laptop running on-line gaming applications tend to exhaust its battery faster than a Smartphone browsing the web. A mains powered desktop PC has no energy concern as such at all.

ENCARA [24] differentiates devices by their energy-related characteristics such as software environment and hardware specification. Consequently devices with critical energy constraints will be protected for longer battery lifetime.

2.2.4 Geographic Routing

Geographic information is a very important piece of the routing puzzle for energy aware path computation. It is more energy efficient for hosts to transmit packets to those closer to them, since it requires less power to perform short range data transmissions. On the other hand it is always efficient to transmit data packets towards the direction of the destination host.

Geographical and Energy Aware Routing (GEAR) [25] makes use of geographical and energy aware neighbor selection algorithms to forward packets to the target region which covers a relatively small number of hosts including the destination. The protocol disseminates the packet in the region by recursive geographic forwarding/restricted flooding mechanisms. First, the area pointing towards the destination is established as a target region which is the preliminary direction for the routing path to follow.

Minimum energy communication network (MECN) [26] is a distributed localized algorithm which computes an energy efficient topology with energy efficient paths from destinations to sources by using each host's location information from its global positioning device (i.e. Global Positioning System).

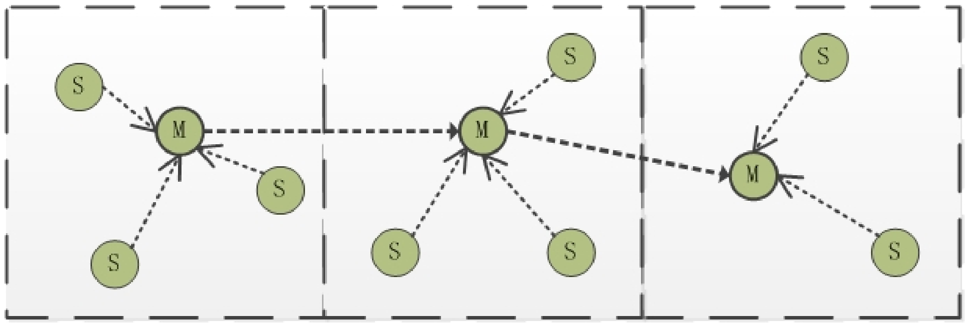

Figure 2.5: Geographic Adaptive Fidelity Protocol

Geographic Adaptive Fidelity Protocol (GAF) [27] makes use of location information gathered from GPS to divide the whole area into virtual grids and each host in the network will use the information to fit in a grid. As illustrated in Fig.2.5, within each grid, the host with highest residual energy will be elected as a master (M nodes) that forwards packets for the rest hosts (S nodes) in the same grid. This solution shares the same idea as hierarchical routing and works with the assistance of location information. The **Power Aware Localized Routing Protocol** (PLR) [28, 28] is a localized routing protocol assuming hosts to know only location information of neighbours and the destination. In this manner, source hosts need only to find the best route to next stop towards the direction of the final destination, rather than finding the optimal route to the destination. PLR needs less routing packets, less overhead and response more quickly to topology change, consequently this mechanism conserves energy for wireless hosts.

Recently, Zhang et al. [29] proposed an **Energy-efficient Beacon-less Geographic Routing protocol (EBGR)**. Conventional geographic routing uses beacon mechanisms to maintain location information for routing decisions leading to extra routing overhead. EBGR does not apply such beacon mechanisms. Instead, a localized routing decision is made when a host has a packet to transmit. First, the host will have to define the ideal position for the next hop based on the direction of the destination and energy efficient distance, and then it chooses the host that is closest to the ideal position with respect to the upper bound set for distance and energy consumption.

Many geographic routing solutions assume the availability of real time geographic information . However, it is challenging to obtain such information in reality, especially in delay/disruption tolerant networks. **Reach-and-Spread**[30] assumes the real time geographic information of mobile destination is unavailable. Therefore it estimates the movement range of the destination based on its historical geographic information. A two-phase mechanism was proposed to reach to the destination range first and spread the message in that range in phase two. The evaluation results show its benefit in terms of delivery ratio, delivery latency and overhead ratio.

2.2.5 Energy Aware Routing with QoS Support

Recently, the networks transport an increasing amount of multimedia content requiring QoS support raising new challenges for the design of routing protocols.

Luo et al. proposed a delay constrained routing algorithm for LEO satellite networks in [31]. The routing of LEO satellite networks is challenging due to the movement of satellites and the network topology. The slow re-routing process results in poor QoS. In this context, the authors proposed a weighted graph model to analyse the routing problem and the proposed routing algorithm uses approximation methodology to find the path that meets the delay constraint.

Sequential Assignment Routing (SAR) [32] constructs multiple trees, rooted at a neighbor, and allows all hosts to have multiple paths to other hosts with different QoS and energy constraints. By estimating the maximum number of packets which can be transmitted before battery depletion and the QoS metric (delay and priority of packets) along a certain route the optimal path can be selected. The QoS metric is evaluated by weighted delay value in conjunction with the priority of packets. Since SAR is designed for Wireless Sensor Networks (WSN) with low mobility, it is table driven and maintained by periodically exchanging energy and QoS updates between hosts.

Real-time Power-Aware Routing (RPAR) [33] protocol supports energy efficient real-time communications in wireless sensor networks (WSNs). It dynamically adapts transmission power and routing decisions based on packet deadlines in order to achieve energy efficiency and delay guarantees for real-time applications. The neighbor management applies on-demand mechanisms rather than periodical beacon transmissions. Consequently neighbor discovery is only triggered when there is no route available, and it only allows hosts that can meet the required bit-rates to reply and join the routing table. In addition, when bit-rate requirements are satisfied, it will decrease the transmission power to improve energy efficiency, otherwise it increases the transmission power to assure effective data transport. This mechanism effectively reduces energy cost for neighbor management. However, RPAR may suffer from sudden congestions.

Bandwidth is an important resource in wireless networks since multiple interfaces may share the available channel bandwidth and cause interference during communication. Therefore, being able to estimate bandwidth availability and allocate it efficiently could be an effective approach to achieve better performance.

EffatParvar et al. [34] proposed a solution which makes hosts query the bandwidth availability of their neighbors and estimate bandwidth requirements for effective communication in order to enable admission control mechanisms. Thus, only effective wireless communications can take place. Furthermore, it encourages local repair of broken links, where hosts try to fix the broken paths or find a bypass locally without initiating new route search.

Taleb et al. [35] proposed a cross layer approach for TCP/RTP-based multimedia applications in heterogeneous wireless environments. In such environments, the bandwidth availability over different network types may vary raising new challenges. To address these issues, the proposed solution makes use of a cross layer approach. A physical layer mechanism detects overlapping access point coverage areas and provides useful information to assist hand-off decisions. The application layer records user mobility pattern and estimates next possible access point. With this support, network layer is able to find alternative (new) routes in the overlapping area. Transport layer uses two routes to setup two connections, of which the new connection uses dummy packets for bandwidth availability estimation via TCP/RTP. This combined solution proved to be efficient in terms of performance and energy efficiency.

REAR [36] is a reliable energy aware routing protocol. It considers residual energy capacity of each host and each candidate route. Hosts will forward routing packets after a delay, which is determined by their remaining energy level. In other words, hosts with high remaining energy capacity will forward routing packets first and will be selected as the next hop. With the depletion of available energy, a path energy level may be lower than a threshold and will be replaced by a new path in order to achieve balanced energy usage. To ensure reliable data communication, REAR applies a retransmission mechanism with acknowledgment packets. Furthermore, it makes use of multiple routes to ensure successful transmission as well.

EDEAR [37] applies reinforcement learning to achieve energy and delay efficient routing. Wireless networks often experience dynamic traffic conditions and this attribute makes QoS support an important requirement for routing. Reinforcement learning process can adjust the routing algorithm based on parameters that reflect current status of the whole environment (i.e. traffic conditions). Therefore, reinforcement learning makes adaptive routing possible, where routing protocol estimates delay and routing cost according to previously collected data. EDEAR collects information regarding the residual energy of each host, energy consumed on each link of one-hop neighbors and link delay from intermediate hosts and compute the cost for each path. EDEAR explores the whole network and uses exploring agents to learn from past experience in order to compute the best path.

DGRAM [38] is a TDMA based energy efficient and delay-aware routing solution. It requires beacon exchange to collect location information and uses slot reuse techniques to achieve shorter delay. First, hosts have to be uniformly distributed and exchange location information to learn the topology. Then, DGRAM constructs a logical structure based on the network topology with several tiers to ensure data packets are transmitted towards the destination from the inner tier to outer tier. Based on mathematical analysis, DGRAM provides delay guaranteed routing with energy efficiency considerations.

EARS [39] is an energy efficient and reliable data-centric routing protocol. Data-centric routing protocols, a widely used type of routing protocols for WSN, suffer from large energy costs resulted from large amount of overhead traffic used to advertise interests and explore data availability. EARS introduces a new metric considering data rate and frame error rate at MAC layer for each host to exclude links of bad quality from the process of advertising interests, so that better communication reliability and energy efficiency are achieved.

2.2.6 Content-based Routing

Content distribution networks (CDN), multimedia content is stored, cached and replicated at server level. User requests for the same content will be directed to several servers in order to avoid server overloading and high delays with negative impact on

QoS. In other words, content-based routing is an act of load balancing for better performance.

An approach to content based routing is to use DNS servers to store information regarding candidate servers for content distribution and redirect user requests to a particular server based on pre-defined performance metrics. Another approach is to make routers content-aware by using upper layer information and construct content-aware networks in order to make routing decisions with respect to content-based policies.

Miura et al. [40] proposed a solution taking the second approach. A natural way to load balance is to aggregate information of each server current load. However, this may result in significant amount of traffic overhead. As a matter of fact, network delay is very important as well and it reflects the condition of links reasonably accurate. Consequently routers are used to monitor each traffic flow and calculate the application layer Round Trip Time (RTT) between the router and the server to record total response time. By doing this, it reduces routing overhead and still delivers good performance. In addition, it applies probabilistic server selection in case client requests concentrate at a single server. However, the latency between the router and the server cannot accurately reflect the latency between the client and the server. Therefore, the closer the client is to the router, the more accurate the results will be.

Hierarchical Content Routing [41] organizes servers with replicated content into clusters to form a hierarchical structure, where routing process is performed intra-cluster-wise and inter-cluster-wise. For intra-cluster content-based routing, hashing-based schemes are applied. Each router maintains a hashing table of content held by other servers within the cluster, so that each router can effectively query another a router for available content with reduced routing overhead. Meanwhile, inter-cluster content based routing is performed using queries. A content request will be forwarded via the cluster head to neighbor clusters. This hierarchical solution is suitable for large-scale scenarios.

Wide area content based routing [42] uses tag mechanisms to support routing. It constructs a virtual content network (VCN) between clients and servers. Once the client request arrives at the edge of the VCN, this routing scheme tags the packet with routing policy information and content information, which will be added to all consequent packets of the same traffic flow. The content aware routers in the VCN will make routing decisions based on the tags. This scheme is efficient and it saves effort for consecutive packets in the same flow and provides another layer for efficient router level data processing.

Carzaniga et al. [43] proposed another content-based routing scheme which is based on message broadcasting. It builds up a virtual layer to include attributes representing content information for packets. Hosts keep track of servers holding certain content by recognizing certain attribute values. When receiving requests, intermediate hosts try to match predicates of requests against their record and find the route to the source.

Table 2.1 presents an overview of the energy efficient solutions for the network protocols discussed in this section.

Table 2.1: Energy efficient Network protocols.

Protocol	**Type**	**Description**
MTTPR [12]	Apply Energy Metric	Select the route with the least total energy consumption.
MBCR [13]	Apply Energy Metric	Select the route with the most remaining battery capacity.
MMBCR [13]	Apply Energy Metric	Improve MCBR by checking remaining battery capacity across devices.
CM-MBCR [13]	Apply Energy Metric	Hybrid solution mixing MTTPR and MBCR.

Protocol	**Type**	**Description**
EAR [15]	Apply Energy Metric	Consider both energy cost of each hop and the residual energy level of each host.
SPAN [16]	Apply Energy Metric	A distributed algorithm in which each node decides to sleep or forward by evaluating the available energy and its contribution to the route.
LEACH [18, 19]	Hierarchical Routing	A cluster based solution that most nodes only communicate with the cluster head, and only the head communicate directly with the base station.
PEGASIS [20, 21]	Hierarchical Routing	A chain based solution that improves LEACH by letting cluster member nodes only communicate with the neighbour instead of the head.
CETAR [23]	Differentiate Devices	Differentiate devices by their major role in the communications: sender or receiver.
ENCARA [24]	Differentiate Devices	Differentiate devices by their energy related characteristics: power source, software and hardware specification.
GEAR [25]	Geographic Aware Routing	Forward packets towards the direction of the destination node.
MECN [26]	Geographic Aware Routing	Use location information provided by GPS to compute the most energy efficient route.
GAF [27]	Geographic Aware Routing	Use location information to divide the topology into grids, and use cluster based mechanism for intra-grid and inter-grid communications.

Protocol	**Type**	**Description**
EBGR [29]	Geographic Aware Routing	Apply beacon less routing, where a localized routing decision is made when a host has a packet to transmit.
Reach-and-Spread [30]	Geographic Aware Routing	Use historical data to estimate the destination location range.
SAR [32]	Energy Aware QoS Support	Construct multiple routes with different QoS and energy constraints.
RPAR [33]	Energy Aware QoS Support	Energy efficient real-time communications: dynamic adaptation based on packet deadlines.
QoS support and local recovery Routing [34]	Energy Aware QoS Support	Make hosts query the bandwidth availability to estimate needed bandwidth.
Cross layer solution for TCP/RTP multimedia [35]	Energy Aware QoS Support	A cross layer solution for TCP/RTP based multimedia applications in heterogeneous wireless networks.
REAR [36]	Energy Aware QoS Support	A reliable energy aware routing. Nodes with more energy forward packets first.
EDEAR [37]	Energy Aware QoS Support	Apply reinforcement learning to achieve energy and delay efficient routing.
DGRAM [38]	Energy Aware QoS Support	A TDMA based energy efficient and delay guaranteed routing solution.

Protocol	Type	Description
EARS [39]	Energy Aware QoS Support	An energy efficient and reliable data-centric routing protocol uses MAC layer detection to reduce unnecessary overhead traffic in data-centric routing.
SPIN [44]	Data-centric Routing	Allow source node to broadcast advertise data before the actual data to improve flooding routing.
Directed Diffusion [45]	Data-centric Routing	Present data as attribute-data pair so that nodes only accept traffic they are interested in.
Routing using passive measurement [40]	Content-based Routing	Make routers content aware and monitor each traffic flow in routers for better performance, energy efficiency and load balance.
HCR [41]	Content-based Routing	Saves energy by organizing servers with replicated content in a hierarchical structure.
WACR [42]	Content-based Routing	Tag packets with corresponding routing policy according to the content, in order to save the effort for consequent packets in the same flow.
Routing for content-based networking [43]	Content-based Routing	A content based routing built on a message broadcasting layer that allows hosts to recognize certain content by recognizing attribute value.

2.3 Cross-layer Approaches to Energy Efficiency

The layered architecture has become the de facto standard for networks and it has been outlined that the success of Internet is mainly determined by its layered architecture [46]. The benefits and efficiency are due to the simplicity brought by modularization. In the layered protocol stack, each layer provides specific services to the upper layer and is dependent only on services provided by the lower layer. The interaction between adjacent layers is independent of implementation details thus making the designing process much easier. This is depicted in Fig.2.6.

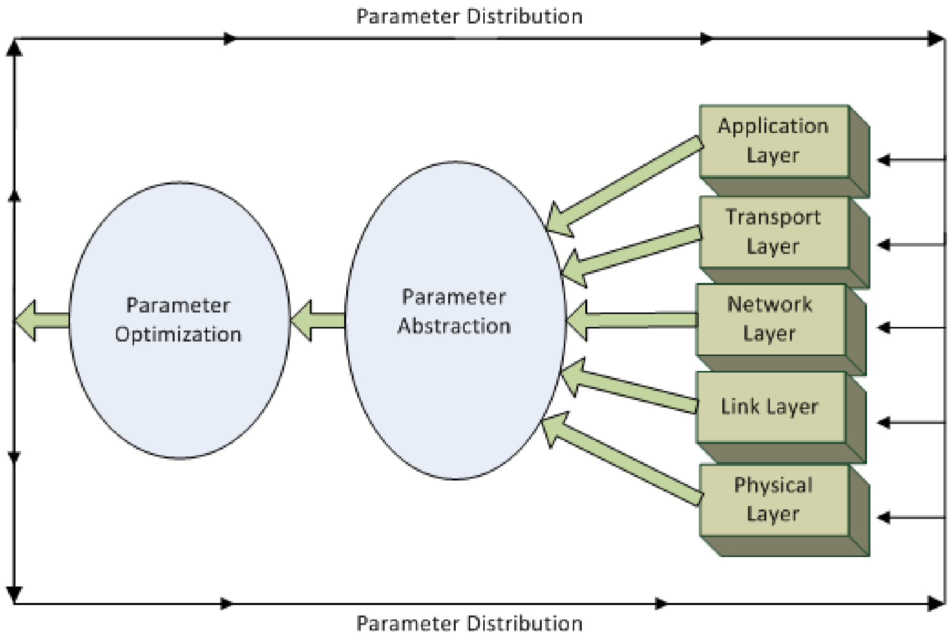

Figure 2.6: Illustration of General Cross-layer Approach.

Despite the wide use of layered architecture, there is an ongoing trend of cross-layer approaches towards increasing the quality of service and energy efficiency. Protocols and algorithms have been widely exploited at each layer, but they are designed in an independent way without consideration of inter-relationships among the protocol stack layers. However, the proper behaviour of each single layer depends on the functionality of other layers especially adjacent layers. Energy saved through modification of a single layer could possibly incur energy waste at another layer (e.g. extra control overhead). On the other hand, the dependence between layers infers that inter-layer information exchange and variables shared across the protocol stack can provide significant benefits to network performance optimization. Therefore, cross-layer approaches are designed and implemented to take a more holistic view of the network protocol stack. Moreover, the unique characteristics of wireless links involves challenges that could not be handled through layered approaches. For example, the congestion control scheme at transport layer could use information from MAC layer to prevent inaccurate detection of congestion situations.

A typical methodology for cross-layer design takes several steps:

- The relationship between layers is identified in terms of both adjacent and non-adjacent layers. The main factors that need to be examined are layer dependencies and data flow through the protocol stack.
- Identify the features of each layer that contributes to the problem to be solved, which in the context of this paper refers to energy efficiency for multimedia content delivery. The contributions are classified into three types [47]: parameters with only local effect on individual layers, controllable parameters having direct effect on the performance of multiple layers and uncontrollable parameters directly affecting multiple layers.
- Optimize the coupling through modification of parameters at each layer. Instead of focusing on a single perspective, cross-layer approach improves the collective performance of several layers which might even sacrifice performance metrics at single layer level.

Although there are many possible benefits of cross-layer approaches, many challenges have to be overcome during the design process. Authors of [48] present the potential issues cross-layer design could cause. It is outlined that fragmentation could be caused by violating the layered structure and there is a high possibility of inadvertent performance loss due to conflicting cross-layer solutions. Energy-efficient cross-layer approaches for multimedia streaming are described in the following subsections.

2.3.1 General Aproaches

Several cross-layer multimedia streaming approaches introduce a parameter abstraction method which absorbs different parameters or policies at several layers of the protocol stack and observes the quality level or energy consumption to compute a parametrised table. This table is used for strategy adoption at each layer towards quality or energy optimization.

A cross-layer solution is proposed in [49] to optimize user experience for video streaming in wireless environments. It mainly involves three steps: first, parameter abstraction at application layer, data link layer and physical layer is generated respectively, then these parameters are integrated to be optimized and finally the optimal values are distributed to each layer. The abstracted parameters introduced by this mechanism avoid trifles and large amount of information that could be gathered at each layer and at the same time hides the technical details. Therefore, this mechanism could be implemented in a more general way and deployed within multiple different systems. Experiments on wireless video streaming have been explored and several parameters (i.e. video source rate at application layer, time slot allocation at data link layer and modulation scheme at physical layer) are optimized for maximum user satisfaction.

The solution proposed in [50] optimizes end-to-end quality of video streaming services in a multi-user environment. It mainly collects status information from application layer, data link layer and physical layer, performs parameter abstraction and optimizes a single objective function through the use of cross-layer parameter tuples.

The strategy proposed in [51] evaluates different strategies adopted by different layers in order to enhance robustness and efficiency of scalable video content streaming. The packet loss ratio and throughput efficiency based on a multi-path model is derived and used to characterize the video distortion model and estimate the channel condition in terms of SNR. Therefore the protocol is able to select FEC at application layer, retransmission strategy at MAC layer and optimal packet size in order to maximize quality and performance.

In contrast to most power saving schemes for infrastructure-based wireless networks, the study presented in [52] introduces an energy efficient mechanism where the decision to transmit packets and to suspend the wireless card is made by the hosts instead of the central access point. There is no beacon scheme in this system, and it is up to the mobile host to start a transmission and to fetch data from the base station. It is also the responsibility of the mobile host to balance the tradeoff between energy efficiency and packet delay. Another feature of this protocol is the decision of switching between modes which is application-driven, meaning different types of applications will have direct impact on the power management policy.

Liu et al. [53] use mathematical analysis to combine energy efficient routing and sleep scheduling in one optimal framework where the mathematical model is provided to give near optimal solution that balances traffic load across the network and periodically switches the WNIC to sleep mode in order to reduce idle listening time.

An effective channel assignment strategy can help to solve medium contention issues. Alicherry et al. [54] propose to combine efficient channel assignment and routing for wireless mesh networks in order to avoid interference and conserve energy. It constructs a formula based on interference constraints, the number of available radios and channels for every router to achieve optimized network throughput and energy usage.

2.3.2 Content Sharing

Content sharing reduces the traffic from the server to the clients. For content sharing, the server transmits the original content to one of the users of a group of users requiring the same content. The content will be then shared within the group of devices. The server reduces the energy consumption to only a fraction of the group size. **GroupDL** [55] is a typical content sharing scheme using the above mentioned mechanism, but it cannot guarantee any energy reduction at the client. In fact, the total energy consumption on the client side is similar with that of traditional solutions. Chen et al. [56] introduce a scheme where devices can request other devices with higher energy levels to download content for them, then transfer it locally. Tests performed on an iPad showed how energy was saved because less 3G interface usage was required for the device with lower energy level. However, the device that performs the download incurs higher energy consumption. Other content sharing based energy efficient solutions [57, 58, 59, 60] take the similar route that offloads content to local wireless network. Yet they have very much the same limitation of not being practical when local networks are often congested, others have lower bandwidth or free AP is not available. Meanwhile the group management and mobility management have always been challenging.

2.3.3 Traffic Shaping

Energy saving can be achieved at MAC layer by maximizing sleep duration of WNIC. A major problem is represented by packets which may get lost if the recipient device is in sleeping mode when they arrive, which means a mobile host needs to wake up frequently to check if there is data arriving. Thus it is more energy efficient to form traffic bursts deliberately at upper layers and inform a host of the next burst schedule to prolong time spent by the WNIC in low power states. A transparent proxy is normally used to receive the continuous streaming from server and form bursts in its buffers, and determine the time to transmit the next burst to the client based on a buffer releasing policy. Burstiness not only helps prolong sleeping interval of WNIC, but is also claimed by several studies [61, 62] to exploit the charge recovery effect with traffic patterns. Experimental results show efficient energy use and longer battery lifetime. The basic mechanism of traffic shaping is shown in Fig. 2.7.

Web traffic oriented protocol, **PAWP** [63], schedules incoming traffic into intervals of high and low communication levels to prolong sleeping interval of WNIC between consecutive data receiving sessions. Data is buffered to form bursts, and data releasing follows several rules: data is released to the client if the WNIC is in low power state; data should be transmitted within a bound interval; data transmission starts if more than a predetermined value of objects are buffered; data forwarding has to be initiated

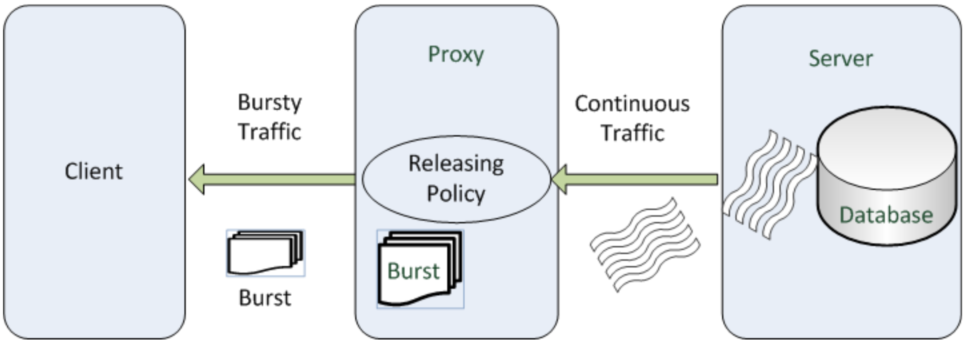

Figure 2.7: Illustration of Traffic Shaping.

whenever the overhead of switching WNIC to active mode is justified.

Catnap [64] is another data oriented protocol which uses the proxy to decide the best opportunity to start data transmission.

Some other traffic shaping mechanisms are designed specifically for streaming applications, such as **PASP** [65] which works similarly to PAWP.

The multimedia oriented mechanism presented in [66], as shown in Fig.2.8, introduces the use of proxies at both the server side and client side. Server side proxies or local proxies mainly shape data traffic generated by the server into bursts and exchanges information, such as the next scheduled data burst, with the client side proxies. The client side proxy will inform the wireless interface about the future traffic so the interface could switch to sleep mode before the arrival of data in order to save energy.

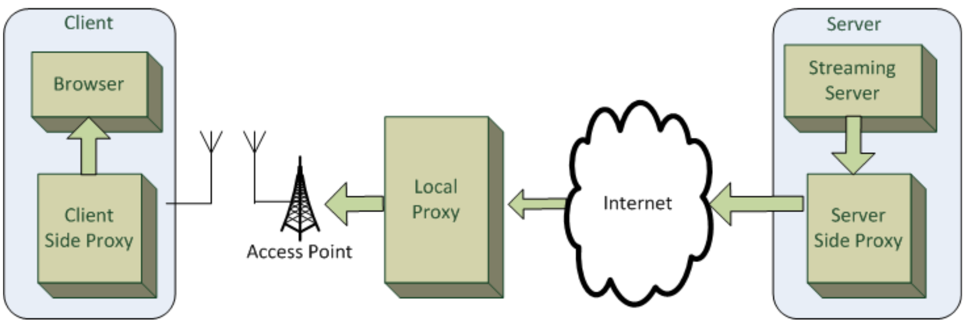

Figure 2.8: Illustration of Application-specific Network Management for Energy-aware Streaming.

Experiment results show up to 83% energy saving as long interval between bursts is guaranteed by traffic shaping. However, traffic shaping using proxies might mislead the behaviour of stations and incurs unnecessary congestion control.

Another proxy-assisted protocol proposed in [67] works in a similar way with the support of more streaming formats. It suggests another approach where the proxy does not provide smoothing function, instead, it sends control packets which indicate the arrival time of next data burst.

In [68], a proxy between client and server is implemented and it is transparent to both sides. Data is buffered at the proxy before sent to the clients and proxies broadcast messages regularly advertising the next round of traffic schedules. The schedule information includes the start time of data transmission to each client and the length of buffered data so the corresponding WNIC is able to wake up before the arrival of the next burst.

The idea of **Multi-modal transport layer** is explored in [69]. It is designed to adapt its behaviour to different environments with the goal to increase battery lifetime. Traffic is shaped in forms of bursts through manipulation of ACK messages. Delayed ACKs result in packet grouping during transmission. The feature of slow start and short traffic transfer, which means traffic bursts are separated by a multiple of RTT, is utilized in the prediction of data arrival time.

2.3.4 Joint Routing and Sleep Scheduling

Another direction in manipulating the behavior of the wireless interface is through the usage of routing information. The main idea behind this is to enable wireless hosts to determine their position within the network. At the same time, routing topology is used to study potential interferences among wireless hosts. This information can be wisely used to achieve efficient utilization of bandwidth resources through allowing simultaneous communication between non-interfering hosts.

PEDAMACS [70] is a TDMA-based protocol with an access point determining which mobile host should occupy a time slot. Fig.2.9 illustrates how PEDAMACS learns the topology from the network layer, and schedules the sleep time slots for nodes to construct an energy efficient route. The first phase of this protocol is to construct a routing tree at the access point by broadcasting topology learning packets. Knowing the network topology, the access point determines when a host will be able to use a particular slot, and the schedule will be broadcast to other hosts.

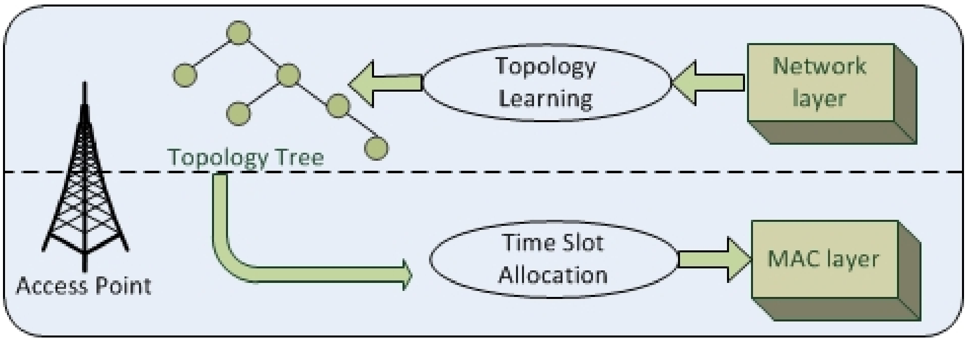

Figure 2.9: PEDAMACS: Joint Routing and MAC Layer Sleep Scheduling.

Other solutions allow mobile hosts to gain information about their position in order to perform self-scheduling. **TDMA-based protocol** presented in [71] assumes a small number of wireless hosts rooted at the sink. Topology awareness could be acquired and the interference patterns among neighboring hosts could be known and controlled. In this scheme, time is divided into epochs, and each host has K slots in an epoch for the purpose of retransmission. A packet is retransmitted for K times at most if not successfully acknowledged by the receiver, thus the delay is controlled and does no exceed the duration of an epoch. Energy efficiency is achieved by assigning different duty cycles for each host according to their position in the predetermined data gathering tree. Routing information is required by each host to construct the data gathering tree.

D-MAC [72] includes a mechanism that builds a data gathering tree which describes the depth of a host within a set of multi-hop paths. An offset is specified as U, and any hosts that resides at the depth of N in the tree will wake up U time units ahead of the destination, which is the top host of the tree. In this case, every host is assigned its own time slot, and the next hop of a path will be able to wake up after the previous hop's host wakes up and sends out the packets. This protocol efficiently saves energy by letting only the hosts on the path wake up, and it also helps reduce packet delay by letting hosts waking up sequentially. However, collisions may still occur when different branches of the same host try to send packets at the same time although a back off scheme is included in this protocol.

In the scheme proposed in [73], each host wakes up when it is time for it to sense the environment and when it expects packets from neighbouring hosts, which means it has to route the packets to the next host. To be specific, if there is a route starting from host A to host C via host B, host A samples and transmits a packet, both of which processes take 5ms, then host B has to wake up 10ms later when host A starts to transmit.

Wu et al. [74] proposed to organize a tree of sensor hosts, which allows data aggregation to be performed along the tree structure in a more energy efficient manner than normal tree structure. Furthermore, from the viewpoint of the MAC layer, sensors consume different energy in different states and during state transitions. This mechanism ensures every host has to wake up only two times in one scheduling period for receiving data from the children and sending data to the parent in order to reduce the frequency of state transition which causes energy waste. This enhanced wake-up scheduling mechanism with the energy efficient data aggregation tree achieves better energy efficiency.

Some joint protocols utilize information from network layer within MAC protocols, while others are intelligent routing protocols which adjust routes dynamically according to host schedules in order to avoid sleeping hosts along the path and decrease end-to-end delay.

Bernardos et al. [75] introduces a TDMA communication scheme allowing neighbor hosts to cooperatively find required communication time slots and avoid redundant message exchange. In this manner, mobile hosts will schedule their wake up time according to predetermined timetables and transmit packets to the neighbours only if both parties are awake prolonging the sleep time. Based on hosts wake up schedules, this routing solution carefully selects the precise path from the source to the destination that includes as many hops as possible in one time frame, so mobile hosts do not have to wait too long before each of them forwards the packets to the next hop in order to achieve shorter delay.

2.3.5 QoS Through Packet Prioritisation

Although quality of experience and energy efficiency are not absolutely opposite to each other, there is normally a trade off between them. High quality multimedia streaming requires reliable data transmission with low delay, low jitter and smooth playback, which implies huge amount of data transmission, high bandwidth and timely response to packet loss. On the contrary, energy conservation is achieved through long period of inactivity, ignoring decrease in performance, or preference of selective hosts, which leads to delayed data transmissions and drop in throughput.

Meanwhile, different applications ask for various types of services. For example, web content tolerates delay, thus could be served with best effort while multimedia streaming requires real-time data transport. In order to utilize limited network resources and balance QoS among applications, researchers have explored the idea of assigning different priorities to packets according to application type, channel condition, energy level and take into account different priorities when making decision on channel contention.

The protocol proposed in [76] integrates the application layer, MAC layer and physical layer in order to maximize QoS. It classifies users in different categories based on both network conditions and associated QoS levels. Different transmission techniques (i.e. channel coding schemes, power control patterns, and priorities) are scheduled among users at both uplink and downlink by the MAC protocol. Different QoS states are defined by user priority/pricing and different characteristics of data traffic.

The prioritisation rules proposed in [77] are based on both their application and MAC layer information, where application type along with the number of hops that a packet has gone through determines the level of emergency. An inter-host scheduling mechanism tries to minimize collision and idle listening in order to achieve energy efficiency by dividing a frame, which represents a round of RTS-CTS-DATA-ACK into contention period (CP) and transmission period (TP) and allocates these period to packets according to their priorities.

The solution proposed in [78] adjusts the contention window according to traffic types and real-time information. Real-time information specifically refers to packet collision rate which is gathered at each host so that packets could be sent in time if channel is expected to be idle and kept in the back-off procedure if channel is busy. Traffic types are differentiated so that real-time traffic such as multimedia streaming is served with higher reliability and quality.

The solution proposed in [79] mainly employs a priority function which is based on frame types, channel conditions, buffer space and multiplexing gain. To be more specific, MPEG video data is divided into three categories: I-frames (intra-coded frames), P-frames (predicted frames) and B-frames (bidirectional frames). Frames belonging to different categories are assigned different priorities. Besides frame time, video streams are given higher priorities if they have better channel conditions. Moreover, users with more empty buffers are given priority as well as those streams that have started data transmissions to gain multiplexing.

The protocol proposed in [80] guarantees QoS in terms of throughput, packet error/loss rate and average delay. At the same time it provides efficient bandwidth utilization by introducing a scheduler which takes consideration of estimated channel conditions at physical layer and the queue status at MAC layer. Decision on the number of time slots allocated to each user is made according to the type of service they benefit of: QoS-guaranteed or best-effort.

Table 2.2 presents an overview of the cross-layer energy saving solutions discussed in this section.

Table 2.2: Energy efficient Cross-layer Approaches.

Protocol	**Type**	**Layers**	**Description**
Application-driven cross-layer optimization [49]	Parameter abstraction	Application layer, data link layer, physical layer	Provides a parameter abstraction and distribution mechanism.
Cross layer optimization [50]	Parameter abstraction	Application layer, data link layer, physical layer	Optimizes a single objective function through the use of cross-layer parameter tuples and parameter abstraction.
Adaptive cross-layer protection strategies [51]	Parameter abstraction	Application layer, MAC layer	Evaluates different strategies and applies optimal parameter configuration at each layer.
Joint routing and sleep scheduling [53]	General cross-layer corporation	Network layer, MAC layer	Balances traffic load for optimal routing and adjusts sleep schedule using mathematical analysis.
PAWP [63]	Traffic shaping	Application layer, MAC layer	Schedules traffic into bursts, which are released according to MAC layer information.

Protocol	**Type**	**Layers**	**Description**
Application-specific network management [66]	Traffic shaping	Application layer, MAC layer	Shapes traffic into bursts and informs clients the arrival of data.
PEDAMACS [70]	Joint Routing and Sleep Scheduling	Network layer, MAC layer	A routing tree is built so that sleep schedule can be obtained at the sender.
D-MAC [72]	Joint Routing and Sleep Scheduling	Network layer, MAC layer	Builds a data gathering tree for time slot assignment.
Cross-layer design for QoS wireless communications [76]	QoS through packet prioritisation	Application layer, MAC layer, physical layer	Differentiate nodes by network condition and QoS, and associates different transmission format according to priorities.
Energy-efficient QoS-aware MAC control [77]	QoS through packet prioritisation	Application layer, MAC layer	Application type and MAC layer information, i.e. the number of hops that a packet has gone through are used to determine the level of emergency.
Cross-layer scheduling with QoS guarantees [80]	QoS through packet prioritisation	MAC layer, physical layer	Utilizes channel condition at physical layer and the queue status at MAC layer to adjust appropriate QoS.

Table 2.3: Energy consumption at architectural level.

Component	Energy Consumption	Energy Efficiency Solutions
CPU	Dominant factor when used by most applications	Lowering CPU speed, e.g. Dynamic Frequency Scaling (DFS).
Display	Dominant consumer for idle system	Reducing the back light brightness.
Wireless Interface	Major energy consumer for network-related activities only	Switching interface off, e.g. Power Saving Mechanism (PSM).
Graphics Card	Important energy consumer only for certain applications, e.g. 3D acceleration.	Circuit level techniques.
Hard Drive	Minor energy consumer.	Placing drives in standby mode, e.g. spin-down technique.

2.4 Energy Efficient Mobile Platforms

Extensive research and development work has been invested in analysing the energy footprint of various architectural components of portable devices, such as laptop PCs, Personal Digital Assistants (PDA) and Smart phones.

Power consumption analysis involves the following modules: CPU, display, graphic card, memory, WLAN card, audio, GPS and GSM (for Smartphones specifically).

All studies targeting this aspect have recognized that the overall energy consumption varies considerably depending on workload and usage patterns.

However, various studies have shown different components of the mobile device's architecture to be the most power hungry. The study presented in [81] shows that the CPU and disk are the most significant power consumers. Experimental results discussed in [82] demonstrate that the graphical display and the CPU are the most power hungry elements while the rest of the components consume significant energy only when being intensively used. Similarly, the study presented in [83] suggests that the graphical display as well as the CPU (depending on the operating frequency) are the major energy consumers, and also the graphic card consumes significant dynamic power.

However, studies published by Intel [84] show that the CPU only consumes very little energy, (only 6% of the overall energy consumed by the system) when compared with other components. According to [8], the graphical display and the GSM modem have been proved to be the most energy hungry components of Smartphone architectures while the memory (RAM) and audio have little effect on the overall energy consumption.

Table 2.3 shows the general analysis of energy consumption and existing solutions towards increasing energy efficiency based on different components of portable devices.

The conclusion which can be drawn by analysing the studies presented above is that energy consumption distribution over the component of a user terminal depends on the type of applications running. More network intensive applications such as multimedia streaming applications will render the network interface as the most energy hungry. In the same manner a graphical intensive applications such as a video game will determine the graphical interface or the processor to be main energy consumer.

However, based on the above studies, there components can still be outlined as the main candidates for the most energy hungry component or sub-system. These are the wireless network interface, the processor and the graphical display.

2.4.1 Energy Efficient Wireless Interfaces

Bluetooth Low Energy (BLE) is an extension of Bluetooth 4.0 which consumes a fraction of the power consumed by the standard Bluetooth enabled products through employing low WNIC duty cycles.

Long Term Evolution (LTE) has a built in feature called Discontinuous Reception

(DRX) which consists of two steps. After each block of data is sent, a timer is initiated. The timer is restarted whenever new data is sent or expires and the wireless transceiver is switched off and enters DRX mode with an (optional) short DRX cycle. A long DRX cycle will be employed if no data is received during the short DRX cycle, otherwise the whole procedure is restarted.

WLAN, also known as Wi-Fi is based on the IEEE 802.11 [85] family of standards. It is one of the main wireless communication technologies designed for portable devices. It supports the built-in power saving mode and the wireless interface can be switched off to save energy on a regular basis.

WIMAX, based on IEEE 802.16 [86] standard, supports power saving mode with QoS constraints. Energy efficiency in IEEE 802.16 is achieved through the employment of sleep or idle modes. In idle mode, the mobile station is not registered with any base station, but it receives downlink traffic through paging. The sleep mode consists of three classes, supporting different quality of service. In power save class one, the sleeping window of the mobile station increases exponentially to achieve maximum energy saving, whereas in power save class two the sleeping window size is fixed. The third power save class is a one-time scheme which means the transceiver of the mobile device sleeps for a predefined period and then returns to the normal mode.

2.4.2 Processors

As the world's largest and second largest producer of microprocessors, Intel and AMD have invested huge efforts in increasing energy efficiency.

Intel has incorporated power management techniques in their processor architectures since Pentium III family.

Intel Centrino mobile technology employ two techniques which can effectively reduce power consumption of processors: Asynchronous Voltage Regulator (VR) Control with Power Status Indicator, and Multiphase Intel Mobile Voltage Positioning

(IMVP) Technology [84, 87].

The latest Core i7 has two important features added as a further step towards energy saving: dedicated Power Control Unit (PCU) and Intel Turbo Boost technology [88]. The first technology enables migration of power management control from hardware to software in order to assist power saving through algorithm updates and better scalability. The Turbo Boost technology dynamically adjusts the core frequency according to real time temperature, power or current in order to decrease power consumption. With the new Ivy Bridge micro-architecture-enabled series of processors, Core i5 and Core i7 are more powerful than all their predecessors in terms of energy efficiency with even better performance [89].

AMD as an important competitor to Intel also seeks energy efficient solutions when building their processors. AMD Cool'n'Quiet 3.0 technology [90] is designed and implemented to improve processor performance including reduced power usage, lower energy costs, and greener PC operation through adjusting processor utilisation in terms of processing speed in order to match performance requirements. Therefore, power consumption could be minimised when there is little demand for high performance.

In the context of mobile devices ARM-based chip architectures are most widely used. ARM processors are significantly more energy efficient than standard processors presented above.

For example ARM uses a technique called Intelligent Energy Management (IEM) which handles system configuration based on the actual or predicted workload. The IEM subsystem consists of the Intelligent Energy Controller (IEC), Dynamic Clock Generator (DCG), System controller and Dynamic Voltage Controller (DVC).

Various studies have approached the issue of energy efficiency of ARM processors.

Real-time applications (such as multimedia content delivery) potentially exhibit variations in their actual execution time and therefore finish earlier than their estimated worst-case execution time. Real-time DVFS techniques exploit these variations in actual workload in order to adjust dynamically voltage and processor frequency to reduce their energy consumption.

The study presented in [91] focuses on the development of a Dynamic Stretch to fit power strategy on an ARM based platform which decreases processor frequency for specific tasks during their execution. Experimental results showed up to 52% energy saving for different execution conditions.

The solution proposed in [92] applies Razor (a hybrid technique for dynamic detection and correction of timing errors) to a 32 bit ARM processor with a micro-architecture design that has balanced pipeline stages with critical memory access and clock-gating enable paths. It obtains 52% power reduction for the overall system at 1 GHz operation through elimination of timing margins.

In [93] measurements performed on a simplified Alpha pipeline showed 33% energy savings by scaling the supply voltage to the point of first failure at extremely low error rates.

In [94], the authors evaluated error-detection circuits on a 3-stage pipeline imitating a microprocessor, using artificially induced voltage droops and obtained 32% throughput gain at same supply voltage (VDD), or 17% VDD reduction (and consequent energy saving) at equal throughput. The authors extended this work to an open-RISC microprocessor core in [95] where in situ error-detecting sequential (EDS) [94], [95] and Tunable Replica Circuits [96] are used in conjunction with micro-architectural recovery support to achieve 41% throughput gain at equal energy or a 22% energy reduction at equal throughput.

2.4.3 Graphical Display

Graphical displays are known to be one of the main power consumers in portable devices [97]. However, in the current climate, even for full size PC monitors or TVs the energy consumption is important as it contributes to global energy demands and the consequent environmental impact.

Currently the most popular technologies for graphical displays include Thin Film Transistor (TFT) also known as active-matrix Liquid Cristal Displays (LCD), Organic Light-Emitting Diode (OLED) and Active-Matrix Organic Light-Emitting Diode (AMOLED).

TFT flat-panel displays use an active-matrix in which each pixel is controlled by one to four transistors. Their characteristics include high colour saturation, high contrast, high speed and good viewing angle. TFTs are widely used for mobile phones displays and computer screens.

In terms of energy consumption, LCDs offer little flexibility as their energy requirements are mostly dominated by the back-light [97].

Organic Light-Emitting Diode (OLED) is a display technology used by television, monitors and mobile phone displays. In contrast with LCDs, the OLED displays use light emitting modules for each pixel and consequently the energy consumption will vary with the type of content displayed.

Active-Matrix Organic Light-Emitting Diode (AMOLED) has been a replacement of traditional LED display for portable device screens and television. It is designed for low-power, low-cost and large-size applications and has been used in newly released mobile phones. However, due to reduced maximum brightness brought by AMOLED, it might be difficult to view in direct sunlight.

The second generation of AMOLED, called Super AMOLED, has been developed and deployed by several mobile manufactures such as Samsung to solve this problem. Samsung's Super AMOLED Plus displays further reduces energy, up to 18% than the old Super AMOLED displays.

The main drawback of OLED is increased energy consumption when displaying white images including white background. As shown in [97] a QVGA OLED may consume 1.0 Watts showing black text on a white background and 0.2 Watts when displaying white text on a black background.

However, the energy variation with content characteristics may be turned into an advantage by using adaptive techniques for content display.

Dong et al. [97] propose a mechanism for estimating the overall display energy consumption depending on the characteristics of the image displayed. This model allows application developers to harvest energy variations with content in order to design the most energy efficient graphical user interfaces (up to 75% energy savings can be achieved in this way according to [97]).

The study presented in [98] explores techniques to minimise the energy consumption of the back-light when a video stream is displayed. The constraint is to minimise the negative impact on user's visual experience. The proposed solution models the problem as a dynamic back-light scaling optimisation problem and proposes an algorithm to be implemented in a cloud-based energy-saving service. Experimental results show how energy savings of 15-49% are achieved on off-the-shelf mobile devices.

Some LCD displays have a zoned back-light arrangement which allows for parts of the display to be turned off or dimmed. The solution proposed in [99] selectively turns off or dims the back-lights for parts of the screen where no object of interest to the user is displayed.

Studies such as [100] analyses user behaviour and propose dynamic adaptation of display operation.

The solution proposed in [101] scales down the supply voltage and reduces the wasted energy caused by the voltage drop across the driver transistor and internal parasitic resistance. The proposed OLED DVS may incur image distortion due to the supply voltage scaling which is compensated by altering the image data based on the human-perceived colour space. However, energy savings of up to 52.5% may be achieved.

Table 2.4 summarises the techniques used to reduce the energy consumed by the display.

Table 2.4: Graphical Display Energy Saving Techniques .

Technique	Method	Application
Usage-based control [100]	Not functional during low-power mode	Interactive applications
Partial display turn off [99]	Disable objects of no interest	Interfaces with active/idle objects (LCD)
Color remapping [97], [101]	Altered look and feel	Graphical User Interface (OLED)
Backlight scaling [98]	Adapts the intensity of the back-light	No restrictions (LCD)

2.4.4 Power Management at Operating System Level

With the increasing demand on low energy consumption for mobile devices and not only, Operating Systems (OS) have included power saving features as built-in components during the system design process. A power manager should be provided by the OS as a mediator between hardware and applications for the purpose of prolonging battery life or reducing the overall system energy footprint.

Microsoft's Windows operating systems have built-in power management features such as system stand by and system component deactivation on idle even from early versions. However, the latest OS from Microsoft, Windows 7 and Windows 8, embed some advanced energy efficiency features.

Windows 7 improves idle efficiency by reducing or even eliminating system background activity. Where periodic activity cannot be avoided due for example to I/O polling requirements Windows 7 implements a timer coalescing feature. This feature allows the OS to synchronize various background activities and to execute them at the same time. This technique keeps the processor idle for longer periods of time. Moreover, Windows 7 defers non-critical background activity when the system is running on battery and also allows services to be started only when specific events occur as opposed to being stated when the system starts.

Latest processor power management (PPM) technologies are also supported by Windows 7 via device driver support. The PPM technologies allow the OS to choose the right processor performance state (power consumption modes) according to the load, and then scale the performance of the system as necessary.

Windows 7 also enables low power modes of various devices such as Adaptive Display Brightness, Smart Network Power, Bluetooth and Low Power Audio.

Windows 8 has been developed to accommodate the growing popularity of mobile computing. Consequently it bring several advanced power management features. These include Metro style application model, idle hygiene, and a new runtime device power-management framework.

Metro style application model suspend background applications in order to save battery power. Idle hygiene increases the time of idle states. Power Engine Plug-In (PEP) is device dependant and increases the idling system time.

Linux-base OS distributions are an increasingly popular option for both full scale personal computers such as Desktop PCs or Laptop PC as well as mobile devices. Based on the design specifications, Linux power management techniques are divided into active and static power management [101].

Active power management refers to techniques used to reduce energy consumption while the system (processor) is in use. Static power management refers to energy saving by entering suspend or low power modes supported by processor, such as standby and sleep modes.

There are two CPU power management frameworks ”cpufreq” and ”cpuidle” supporter by Linux. The ”cpufreq” framework allows the CPU clock speed to be adjusted and the ”cpuidle” framework allows the processor to go into a sleep mode when the CPU is idle.

The ”pm-qos” framework provides power management in the context of quality of service requirements. This allows subsystem,applications and drivers to register their performance requirements.

IEEE 802.11 MAC layer support the ”pm-qos” framework and has the ability to bring the wireless interface into sleep mode for a certain amount of time when no traffic is detected. The access points support this by buffering their frames while the interfaces are in a low power mode.

Android is an open source mobile device operating system developed by Google. It is based on Linux kernel but with more functionality added. Its power management takes a more aggressive manner to conserve energy than standard Linux Power Management (PM). A core power driver is added to the Linux kernel in order to assist the PM controlled peripherals including screen display and backlight, keyboard backlight and button backlight [102]. It adopts the idea of wake locks used by applications to request CPU resources, and the CPU is shut down if no active wake lock is detected indicating no service/applications are requesting processing power.

Symbian, as an open OS, has a built in power management in its kernel as well. The power manager will shut down the peripherals that are not in use to conserve energy. Another feature of Symbian OS is that it uses less memory which leads to less energy consumption. The operating system is based on symmetric multiprocessing (SMP), which provides power management through varying the CPU performance by frequency scaling or by switching CPU cores off when not being used. Power Saving Mode (PSM) is introduced in Symbian 3 to further assist prolonging battery life. Processes running on the device can switch to PSM according to their own logic. Moreover, a set of services are disabled when PSM is turned on including Bluetooth, WLAN scanning, display settings, tactile feedback, vibration, 3G/2G, keypad tones, audio feedback.

Palm OS is a mobile operating system initially developed by Palm for personal digital assistant (PDA) and has been extended later for Smartphones. It basically supports three operations modes: running mode, doze mode and sleep mode. The power manager is event driven, and running mode is entered to process any user event. Once the last event is finished, the device enters doze mode. While in doze mode, no operations are performed but the system interacts with the user. Sleep mode is initiated either when the device stays inactive for a predefined interval or when the ”off” button is pressed by the user. During this stage, the system shows off status while the real-time clock and interrupt generation circuitry are still running in the background.

The Apple OS (iOS) is derived from Mac OS X to suit the requirements of mobile devices. It is designed only for Apple hardware products. Due to security reasons, only a limited set of system frameworks and applications have access to the kernel, and a device-level class instance is obtained through the application layer to provide interface to battery information. The power management of iOS is a generic energy conservation mechanism and depends significantly on individual applications. However, it supports the basic function of switching off hardware subsystems which are not in use in order to save energy.

BlackBerry OS is an operating system developed by Research In Motion (RIM) and utilized by BlackBerry products only. It provides an interface which allows applications to capture power status notifications. Similar to iOS, it prohibits direct access to kernel-level power management. However, some client-side add-ons are provided to assist power management. For example, BatteryStatus is a history-based tool to predict battery life in terms of remaining usage time. In the latest version of BlackBerry OS, a battery saving mode is introduced to allow users to configure at what battery level the high efficiency operation mode should be triggered in order to prolong battery life.

Table 2.5 presents an overview of the power management features implemented in mobile OS distributions.

Table 2.5: Energy saving technologies at OS level.

	Company	**License**	**Power Management Overview**
Windows 7	Microsoft	Proprietary	Idle Efficiency, Timer Coalescing, Processor Power Management, Device Low Power Modes.
Windows 8	Microsoft	Proprietary	Metro Style Applications, Idle Hygiene, Power Engine Plug-In.
Windows Phone	Microsoft	Proprietary	Eight different states provided: On, BacklightOff, ScreenOff, UserIdle, Unattended, Resuming, Suspended, Off.
Linux	Linux Comunity	Open source	CPU clock speed adjustment and QoS support. Low power state
Android	Google	Open source	Core power driver added to control peripherals. Wake locks used for request and release of CPU resource.
Symbian	Symbian Foundation	Open source	Peripherals controlled by power manager. Process-based power saving mode introduced. CPU performance adjusted by SMP.

	Company	License	Power Management Overview
Palm OS	ACCESS Systems Americas	Proprietary	Three modes supported: running, doze, sleep.
iOS	Apple	Proprietary	Hardware management through kernel. Switching of power saving mode by third party allowed.
BlackBerry OS	RIM	Proprietary	Hardware management through kernel. Power management add-on provided.

2.4.5 Background Works

This subsection discusses previous effort in terms of energy modelling for mobile devices.

2.4.5.1 Measurement-based Energy Characterisation

Margi et al. [103] proposed a task-level energy characterisation for laptop computers. This is an early attempt to measure the task-level energy consumption, such as processing, I/O access and data transmission, in order to characterising the energy consumption for the whole system. An extensive test for mobile phone energy characterisation is presented in [104]. Measurement-based modelling for mobile platforms can also be found in [8, 9]. The authors of these works conduct a series of tasks and measure the typical power of the whole system for each task, including texting, audio play out and web browsing. Sharing their results from some typical tasks of device usage, these works made the contribution of helping other researchers to gain an intuitive and straightforward understanding of the device energy characteristics. However, the above mentioned research and other research alike fail to provide a generalised accurate paradigm of energy modelling for arbitrary devices running arbitrary tasks.

2.4.5.2 Utilisation-based Energy Modelling

In order to provide a generalised accurate paradigm of energy modelling for arbitrary devices, utilisation-based energy modelling is proposed by some researchers. They all keep record of the utilisation of subsystems, such as CPU, graphics card, WiFi and so on, and the corresponding system power. Linear regression and other techniques are used to explore the relation between the utilisation values and system power for the sack of formulating a generalised device energy model. The formulated energy model will be used for further energy estimation. The real system power readings need to be recorded to compare with the calculated power during linear regression training phase. Different techniques of obtaining the real system power are used, including using external hardware, using operating system battery interface and using knowledge of specific hardware or operating system. They all have their advantages and disadvantages: more accurate modelling often need specific knowledge and external hardware. Hence the more accurate modelling techniques fall short in terms of generalisation and practicality.

Zhang et al. [105] proposes PowerBooter, a power model uses battery-dependent discharge curve and device built in voltage sensors for power estimation. They suggest the model training should include all relevant system states/parameters rather than the typical states in daily device use. On the contrary, other mentioned research only train the model with typical available applications instead of arbitrary loading application that loads the system as variously as possible. The authors take a discharge curve as a reference for real power readings in model training. The concurrent work from Dong and Zhong [106] proposed a automatic construction of a power model using a smart battery interface. This technique requires knowledge of the discharging current and remaining battery capacity, which are not available for most phones. However PowerBooter needs to obtain the battery discharge curve from the current phone, which will have a different discharge shape as time passes.

There are limitations for utilisation based modelling. First, modules like GPS and camera have no quantitative utilisation states. Secondly some system calls that can change power states, for example, file I/O operations, do not provide quantitative utilisation. Last, in Windows Phone, GPS and NIC have ”tail power states” which affects power consumption even after they are powered off. To optimise the system energy efficiency, the modern mobile OSs use smart power management strategy to automatically adjusts system behaviours. These behaviours are easily caught by tracing the corresponding system calls. Eprof [107, 108] was proposed to address these issues.

Eprof is a system-call-based solution that monitors both utilisation-based and non-utilisation-based power behaviour via the available system calls from mobile operating systems. The existing power modelling techniques that take only module usages for estimation have flows. The authors claim Eprof delivers better accuracy as utilisation-based solutions' periodic update imposes delay in reading the change of components with simple binary states: on and off. Secondly they observed the system calls of opening/closing files and sockets on Windows phone do not change utilisation much, yet cause more pronounced change in power states. Since system calls are used as triggers of state change, a Finite State Machine (FSM) is used. It constructs a FSM for each sub system, and combine them together to form a unified state machine for the whole system.

However, PowerBooter and Eprof need an external power meter and OS kernel specific knowledge (e.g.ordering of system call) to construct the model. This approach severely compromises their generalisation in practical. This is because the major OS distributors (Apple, Android) frequently release their OS of newer versions with improved kernels. Obviously an external meter is only possible in lab settings. Moreover, the merit of Eprof is only very much pronounced in Windows Mobile Operating Systems. For utilisation-based subsystem, it also uses regression to build the utilisation-based estimation model. Hence the contribution is to use FSM to deliver better result for non-utilisation-based subsystem.

Sesame [106] provides fine-grain (less than one second) self modelling where no external assistance from outside of the mobile system is needed. The laptop and smartphone oriented solution uses the battery interface already available from the mobile systems. Sesame is claimed to be superior in three aspects: it has higher rate; it uses dynamique predictors instead of manual examination of the system power characteristics; PowerBooter needs battery discharge curve while Sesame using battery interface directly. This solution is tested on a Thinkpad T61 laptop, whose power is measured by a USB data acquisition system. From tests with various settings on laptop and mobile phones, they observe that hardware configuration, usage and battery suggest the needs of self-built modelling solution. Specifically, Sesame uses predictors accesible from OS file system. It runs computation intensive tasks only when the device is powered by *AC charger*. Theoretically Sesame could select any predictor, the location and name of the corresponding files are system specific though. In fact, the authors use general distribution of Linux on laptop for the proof of concept. Hence Sesame fall short when applied for mobile devices. In addition, Sesame developed a dedicated system call that is platform-specific. It imposes more overhead as it reads battery interface readings. One observation is the accuracy is higher when using average battery interface readings. This will incur less readings for training. Consequently the generalisation of the produced model is compromised. The authors suggest to offload model training to the cloud and keeps several models for different system configurations. They observed the accuracy is low at high rate: due to overhead of data collection; non linearity of the relation between power and predictors.

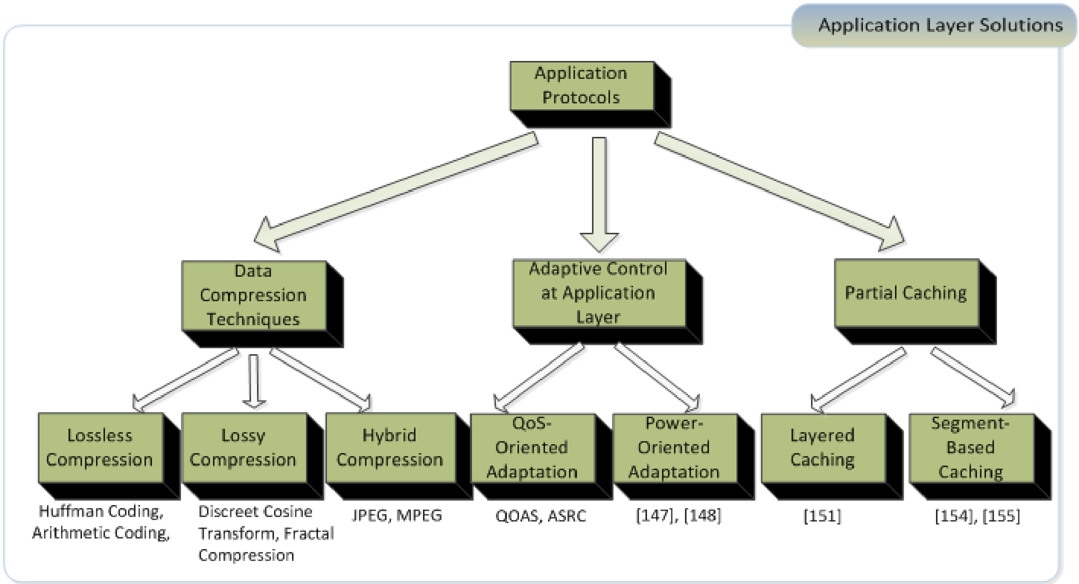

Figure 2.10: Classification of Application Level Optimization Techniques.

2.4.6 Energy Efficient Applications

Application layer protocols provide host-to-host communication and service requirements for various applications. Energy saving at this layer is more application specific and varies with different requirements, especially QoS constraints. For example, web traffic is more delay tolerant, while multimedia applications require real-time communication services and tolerate a certain level of packet loss. Application level energy conservation techniques are categorized based on several aspects including data compression, adaptive control at application layer, data caching, and content sharing.

The first approach, compresses data to decrease its size for transmission and storage efficiency. The second one adapts the behaviour of applications at runtime according to QoS or energy levels. Data caching schemes store a part of potentially highly demanded data in order to increase response time. The key idea of load partitioning is to migrate the computational pressure from the client device to the server. This is intensively studied in various papers including [109, 110, 111]. However, load partitioning is not widely used by multimedia streaming applications as they do not necessarily require intensive computation at the client side.

The classification of application level QoS and energy control for multimedia streaming is depicted in Fig. 2.10.

2.4.6.1 Data Compression Techniques

There is significant research effort targeting data compression techniques mainly due to the limited network resources that have to be used to convey multimedia traffic. The main goal is to compress the large size multimedia content into smaller size encoded content. Compression affects the energy consumption in two aspects: processing power for compression/decompression and power for data to be transported.

General compression techniques such as those presented in [112, 113, 114] have been widely deployed. Techniques have also been developed for specific types of content such as text and database queries [115, 116].

Among all applications, multimedia streaming ones use compression the most due to:

- Scarce power resources: energy efficiency is critical for battery powered devices, and consequently the amount of data transmitted has to be kept as low as possible.
- Large quantity of data required by images, audio and video content: in contrast to text based applications, multimedia content generates vast amounts of data.
- Relatively low bandwidth available in wireless network environments: although there is an ongoing progress in the development of high bandwidth wireless technologies, the increasing growth in wireless users and data traffic is much faster than the new technology development and deployment.
- High QoS requirements: multimedia applications, especially audio and video streaming, are more sensitive to delay and packet loss, which puts higher pressure on network performance.

Multimedia compression mechanisms are used to reduce the amount of data either stored at the sender or delivered to a destination node which recovers the original data through decompression. The energy cost related to the whole process includes the energy consumed with compression/decompression and data transmission. Although energy spent with communication has decreased for the wireless network interface due to the reduced data size and shorter duration of data reception, the technique introduces extra computational overhead, especially in the decompression process at the recipient side. Therefore it is crucial to balance these two aspects in the design of compression mechanisms.

Lossless compression techniques such as **Huffman coding** [117], **Arithmetic coding** [118] or those presented in [113] and, [119] allow the receiver to recover entirely the original uncompressed content. Usually these techniques are not very efficient in terms of the size of the compressed stream. The general principle uses character mapping into binary codes through the use of a mapping table.

Other techniques with higher compression rates provide lossy compression by discarding data deliberately and are more practical for multimedia streaming due to higher compression efficiency. Transform-based techniques translate data into another mathematical domain for data compression. Among the most popular transforms are **Discreet Cosine Transform** (DCT) [120], **Fractal compression** [121] and Wavelets transform.

Prediction-based solutions [122, 123] encode only the difference between consecutive data sets and are mostly applied in audio streaming applications where signals change smoothly.

Another type of layered-coding based techniques prioritizes data into different tiers and apply different compression technique or parameters based on the priorities. Such techniques include those presented in [124], [125] and [126].

Vector quantization is another mechanism used by compression techniques such as **Code-Excited Linear Prediction** (CELP) [127]. The principle behind this scheme is to divide large vectors into groups and represent each group by a codeword.

The energy optimization technique proposed in [128] consists of two parts. The first part adapts the encoding rate of each frame to the degree of motion activity. The adaptive rate scheme means encoding rate is decreased by skipping redundant frames when motion level is low without degrading video quality seriously.

Two techniques are adopted for frame-skipping: direct comparison and forward prediction. The first approach compares the current image with newly captured images and does not encode a frame until the comparison results show violation of quality or latency constraints. Consequently the quality requirements will be met with the least number of frames encoded. The main disadvantage of this technique lies in the fact that a look-ahead buffer is required for comparison and an encoding of a frame is delayed until the next frame is selected. On the other hand, forward prediction adjusts the frame skipping interval based on the difference between the current frame and the previous images. The second approach is to buffer the input data in order to delay data processing which results in longer slack time between consecutive processing tasks and thus achieves energy efficiency.

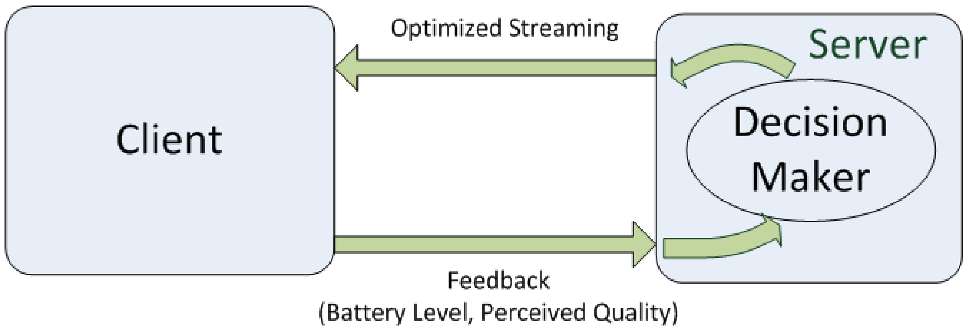

Figure 2.11: Illustration of Adaptive Control at Application Level.

Hybrid solutions [129] make use of both techniques in order to achieve higher efficiency and better quality.

Photographic Experts Group (JPEG) [130] was designed for image processing and provides both lossless and lossy compression.

Studies have demonstrated the high energy efficiency achieved by the technique proposed in [131].

Various algorithms for video compression have been developed and standardized and are widely used today. Some are in the public domain and were developed by organizations such as International Organization for Standardization (ISO), International Electro-technical Commission (IEC), and International Telecommunication Union (ITU).

Moving Picture Experts Group (MPEG) group has developed several standards for moving pictures and audio compression. MPEG-1 and MPEG-2 [132, 133] standardize compression of full motion video. This standards utilize the difference between frames in the process of compression.

MPEG-4 consists of two distinct compression algorithms: MPEG-4 Part 2 (Visual) [134] and MPEG-4 Part 10 (AVC) [135]. Although built on similar principles, MPEG-4 offers much higher flexibility than the previous MPEG standards.

The solution described in [136] provides higher compression ratios and treats video as objects which could be handled both individually and collectively.

The algorithm presented in [137] introduces XML for description of content and supports a broader range of applications [138].

2.4.6.2 Adaptive Control at Application Layer

In order to optimize energy efficiency and improve user experience, applications can be designed to dynamically adapt the streaming process to network conditions, observed power levels or user-side quality measures, as shown in Fig.2.11.

Recent studies have been focused on interactive compression techniques where the sender adaptively chooses the optimal compression rate to meet energy efficiency or QoS constraints.

The solution presented in [139] adapts the compression strategy based on feedback from clients and guarantees the constrained minimum QoS. The battery-powered client device sends the server its maximum decoding capability, so the server could calculate the optimal transmission rate. It is suggested that if the decoding aptitude (P) matches the number of correctly received packets at the client side (Q), the client will achieve the best energy efficiency. If the client receives more packets than it can decode in real time, the energy spent on receiving Q-P packets is wasted. On the other hand if Q is smaller than P, the server should send more packets to improve video quality. Thus the feedback mechanism helps balance these two values and achieve energy efficiency.

Quality-Oriented Adaptation Scheme (QOAS) proposed in [140] uses feedback received from clients, which mainly includes user perceived quality and QoS parameters such as average loss rate, to dynamically adjust the streaming rate at the server side. Simulation results show significant increase in the number of clients that can be served simultaneously and meanwhile the quality of service is maintained at high level.

The mechanism proposed in [141] selects the optimal image compression parameters at runtime to best balance the tradeoff between energy, latency and image quality. The methodology consists of two steps. In the first step, the average value of the image quality and latency is calculated. In the second step, the data obtained from the first step is used to generate a table with quality and latency constraints and total energy consumption spent on computing and transmitting images. The table is then used to look up the optimal parameters for the desired energy/latency/image quality.

Adaptive Source Rate Control (ASRC) [142] for video streaming applications is proposed to work with hybrid Automatic Repeat Request (ARQ). It takes advantage of high throughput and reliability achieved by ARQ and at the same time guarantees that data could get to the destination within the delay limit. ACK packets received at the data source are used to calculate packet error rate which is an indicator of channel condition. the ASRC scheme forecasts the channel effective data rate based on the error rate before the next video frame is encoded. Finally the target number of bits for the next frame is calculated according to the channel conditions and the target delay limit so that the packets can be transmitted correctly and within the imposed delay limits.

The solution presented in [143] adjusts video streaming strategy at runtime to prolong service time of a whole wireless communication system. The authors observed that the video quality is determined by three aspects: encoding aptitude of the server, decoding aptitude of the client, and the channel. Therefore they propose a strategy where transmission power level at the server side and decoding scheme at the client side is adjusted at each frame based on the energy level at runtime with guaranteed minimum video quality. The adjustment is made with consideration of energy level on both sides as the system life time is maximized if the server and client run out of energy at the same time.

EVAN [144] and **ESTREL** [145] use this approach, adapting the video quality based on device characteristics and remaining battery levels. Scalable video coding such as MPEG-4 SVC [146] enables layer-based multimedia quality adjustments. Devices subscribe to enhancement layers only if their remaining energy levels are high. Otherwise they un-subscribe from some enhancement layers to reduce the amount of data to be received/transmitted and save energy. **SAMMy** [147] is a dynamic video delivery solution that adjusts content quality based on estimated signal strength and monitored packet loss rate. These parameters are utilised to make more efficient use of the wireless network resources, increase user perceived quality and save energy. **DEAS** [148] adaptively changes the video QoS level by monitoring the application holding on and the current residual energy. DEAS is the first adaptive streaming solution that considers application running environment (i.e. not only the current multimedia streaming application, but also other applications) and device features that put different energy constraints on the device. Alt et al. have proposed [149] that assess the level of movement between continuous frames. The frames with major difference than the previous frame have to be delivered as important information is lost otherwise. However it drops frames with little difference in movement to save energy. Park et al. [150] have proposed a SNR scalable architecture that trans-coding from H.264 to SVC for energy saving. They developed a dedicated chip for trans-coding in order to release mobile CPU from the computational complexity of trans-coding in adaptive content delivery scenario.

The power-aware scheme introduced in [151] dynamically adapts the behaviour of applications according to energy levels in order to prolong battery life for mobile devices. High quality of service is achieved if the battery resource is plentiful. Energy conservation is performed at the expense of user experience when a device is running out of energy. Experiments based on four different application types: video player, speech recognizer, map viewer and web browser are performed and testing results show that lowering data fidelity yields significant energy savings.

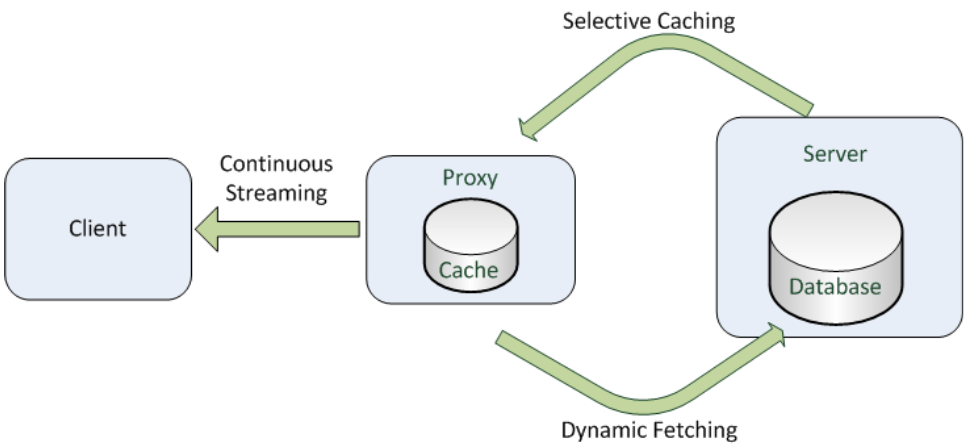

Figure 2.12: Illustration of Partial Caching.

2.4.6.3 Partial Caching

Caching techniques are used in many applications to improve user experience as they can significantly decrease access delay.

Web caching [152, 153] has been explored by many researchers as web content normally is small in size and relatively static. Different from traditional web browsing or data downloading, multimedia streaming, which is increasingly popular nowadays, has tremendous bandwidth requirements and expected user quality of experience. Download before watching solutions are suggested and employed by some protocols. However it is not feasible in many cases as it requires huge amount of buffer space at the client side, and a significant amount of startup delay is introduced. Therefore partial caching of data using proxies is employed in order to perform high quality streaming. The mechanism is depicted in Fig. 2.12.

One partial caching technique performs layered caching based on prioritisation of specific frames. A layered approach is considered in [154] where each stream is distributed into different layers with distinct priorities. The layering mechanism caches the base layer for each stream and discards the last segments of the least popular layer. It performs quality adaptation to the variation of client bandwidth and allows the average quality of a stream to be proportional to its popularity and the quality variation be inversely proportional to its popularity in order to finally achieve efficient cache state.

Other approaches perform segment-based caching where objects are divided into segments and only a small part of them are cached to decrease startup delay. The other part are fetched on demand.

A **prefix caching approach** is proposed in [155]. It stores a large number of initial frames in the buffer and retrieves the remaining frames when a streaming request is made by the user. The size of the cached data depends on both the physical constraints, for example bandwidth and transmission distance, and the quality required by the user such as the maximum playback delay.

The mechanism presented in [156] works in a similar manner. It also takes into consideration the popularity of the multimedia content when deciding whether to replace a video or to cache a video, however, the quality of the cached video is also taken into consideration. When the popularity of a video stream increases, the quality and size of the cached frames increases.

The solution proposed in [157] provides high QoS levels and achieves high resource efficiency at proxies. It mainly consists of two parts: adaptive and lazy segmentation, and active pre-fetching. The adaptive and lazy segmentation adapts the segmentation function to user behaviors and tries to segment the video object as late as possible by adopting three functions. The first part is aggressive admission policy which caches the whole media object when it is accessed for the first time based on the assumption that the future access behavior of a new object is unknown at the first access. An object is segmented adaptively according to the average client access length computed at runtime instead of before access. The two-phase iterative replacement policy will decide the candidate segments to be replaced based on the average number of requests, the average duration of access, the length of cached data and the predicted future access probability which is calculated based on the average intervals between requests and the current time. Continuous streaming is guaranteed by calculating the start point of pre-fetching based on the streaming rate and pre-fetching delay at runtime. Other similar protocols use variable size segmentation technique.

The solution proposed in [158], divides the video stream into variable-sized segments which are assigned different priorities for caching and replacement. The segmentation policy is based on the distance between a segment and the start point of the whole stream, which means the closer a segment is to the beginning frame, the smaller it will be. The reason behind this is to achieve high buffer utilization by discarding a large size of data chunk when replacement is performed. The replacement policy depends on the popularity of a segment and the distance to the first video frame, where the popular segment and the first segments are treated preferentially.

The mechanism used in [159] combines both uniformly sized segments and variable sized segments when dividing video streams.

Table 2.6 presents an overview of the energy efficient solutions for networked applications discussed in this section.

Table 2.6: Energy Efficiency at Application Layer.

Solution	**Type**	**Energy Impact**
Huffman coding [117]	Lossless compression	Processor and Network
Arithmetic coding [118]]	Lossless compression	Processor and Network
DCT [120]	Lossy compression	Processor and Network
Fractal compression [121]	Lossy compression	Processor and Network
Delta modulation [122, 123]	Lossy compression	Processor and Network
CELP [127]	Lossy compression	Processor and Network
JPEG [130]	Lossless and lossy compression	Processor and Network.
Energy-aware MPEG-4 FGS [139]	QoS-oriented adaptation	Network
QOAS [140]	QoS-oriented adaptation	Network
ASRC [142]	QoS-oriented adaptation	Network
Energy-aware video streaming [143]	Power-aware adaptation	Processor and Network
Energy-aware adaptation [151]	Power-aware adaptation	Network
Multimedia proxy caching mechanism [154]	Layered-caching	Network
Proxy prefix caching [155]	Segment-based caching	Network

Protocol	Type	Energy Impact
Proxy caching mechanism [156]	Segment-based caching	Network
Segment-based proxy caching [158]	Segment-based caching	Network

2.5 Chapter Summary

This chapter starts with the introduction of the layered protocol model and indicate how the energy conserving mechanism can be applied to each layer. Since the major contributions included in this thesis, i.e. energy efficient routing and adaptive delivery, mainly concern Network Layer and Applications Layer, the state-of-the-art of the energy efficient solutions on these two layers are classified and explained with details. Besides, the major cross layer proposals are also described. Nowadays energy efficiency of mobile devices regarding both hardware and software solutions is of great importance. Therefore this research on energy efficiency concerns the heterogeneity and differentiation of mobile devices. Notably, different types of energy efficient mechanisms on mobile devices are discussed in this chapter. A better understanding of mobile devices and the existing research on different layers plays a crucial role during my work.

Chapter 3

Energy-oriented Application-based System Profiling and Prototype-based Testing

3.1 Overview

This thesis presents the Energy-oriented Application-based System Profiling (ASP) for mobile devices that: a) constructs a mapping between the work load on each hardware component and the corresponding system power consumption value; b) monitors the work load on the hardware components for each application, in order to construct an application profile table that records the power signature of applications; c) calculates the energy constraint of current context by referencing the above tables and monitoring current status of the device (screen brightness, application type, etc).

As the output of ASP, the energy constraint information is used for Energy Efficient Routing and Energy Efficient Adaptive Delivery, which are presented in the following sections. The routing protocol and the cross layer adaptive delivery protocol are considered users/consumers of ASP.

ASP is performed in three phases: setup, monitoring and update. ASP focuses on those components of the latest mobile devices [9] which are the major energy consumers among all the hardware components. These are screen (SCR), graphics processor (GRA), WLAN interface card (WLAN), such as WiFi for example, cellular network interface module (CELL), such as GSM, UMTS, LTE, etc. and processing chip set (CPU).

The *ASP* is implemented in a prototype system dedicated to work for smart mobile devices, including smart phones and tablet PCs. The system adopts a pure software application level approach, where no modification on hardware and software network protocol stack or external devices is needed. The testing results are obtained on a Samsung Galaxy Note 10.1 tablet with the ASP prototype system installed.

This proposed model is the first application-aware dedicated energy modelling software for mobile devices, smart phone and tablets, that has its design, online model construction and testing completely performed on the mobile device itself instead of lab settings. The conducted comprehensive real tests are the first to target tablet devices. In fact, Samsung Galaxy Note 10.1 is used throughout for the design, software development and all the tests. In addition, this is the first solution to feature comprehensive application aware modelling. It also features incremental model updating.

First, a regression based energy-oriented system profiling system is built. This system runs a series of tests to record battery discharge while subsystems are loaded at different levels in order to build a machine learning based device energy model. This energy model takes hardware subsystem usage as input and calculates the estimated energy constraint score.

During daily device usage, the profiling system keeps record of the power signature of individual applications by periodically obtaining and recording device parameters. Referencing current application type and the energy model, this system will give the score of current device energy constraint on demand in real time. Notably, the energy model construction takes incremental training approach that updates the energy model periodically to suit the latest device usage scenarios, including user habit, hardware worn and so on.

The use of ASP imposes overhead to the system. Firstly, it needs periodically sampling to collect system utilisation data. Secondly, the complex calculation is needed to train the device energy model and calculate device energy constraints. To reduce overhead, ASP keeps and uses application power signature to reduce the frequency of sampling data from the system. Besides ASP trains its device energy model while device is charging.

The data collection overhead is computed as File Input Output cost * Sampling Rate. The File Input Output cost is hardware related, and consequently real-life measurement is needed to quantify the data collection overhead. Similarly real-life measurement is also needed to quantify the overhead of model training. In this context, the total overhead was evaluated on the prototype system as illustrated in the subsection Overhead Analysis.

3.2 Contributions and Advantages

ASP include features from several existing energy modelling techniques. As in Sesame [106], the battery interface is used instead of external hardware. Battery interface is a set of Linux API functions which provides all the information regarding battery status, including voltage, current, and capacity, etc. This enables self constructive software based device independent energy modelling. This has two benefits, practicality and device independence. For the incremental model update, the training only happens when the device is charging to minimise the energy consumption on modelling. As PowerBooter [105], I train the model using many variations of loads rather than some typical loads for several existing applications. This effort is to obtain a generalised model that suits many potential new applications.

Compared with the above existing energy modelling techniques, There are new observations made during the field tests. More importantly, ASP addresses some problems which were not addressed so far.

- Many papers explored the construction of a power model for mobile-phones, but as far as known, this is the first model constructed for a tablet device. Although these models are similar in many aspects, such a model is important since it permits the analysis of the energy is being spent in the tablet, and once the model is created, the power estimation can be done with a higher sampling rate and accuracy rate compared to the battery interfaces of the tablet.
- Application power signature is proposed to reduce monitoring overhead. This is because any individual application shows distinctive energy characteristics on different device subsystems.
- The proposed model does not rely on any external power measuring equipment/tool to validate the model. This is because external tools are not available in daily device usage, and most likely cannot be used for model updating and further training.
- The energy modelling technique adopts incremental updating in order to adapt the model to the change of battery and user preferences.
- Field tests demonstrate the difference between LCD and OLED in terms of energy characteristics complementing the many works done on OLED, a realistic yet different energy model for LCD, is believed to be used for big screens, particularly tablet screens.
- ASP features a better refined CPU energy model and addresses issues that are not fully understood in the existing works. For instance, the relationship between CPU load and CPU energy usage was not comprehended by existing researches [5]. ASP takes into account that modern mobile CPU features multiple cores and

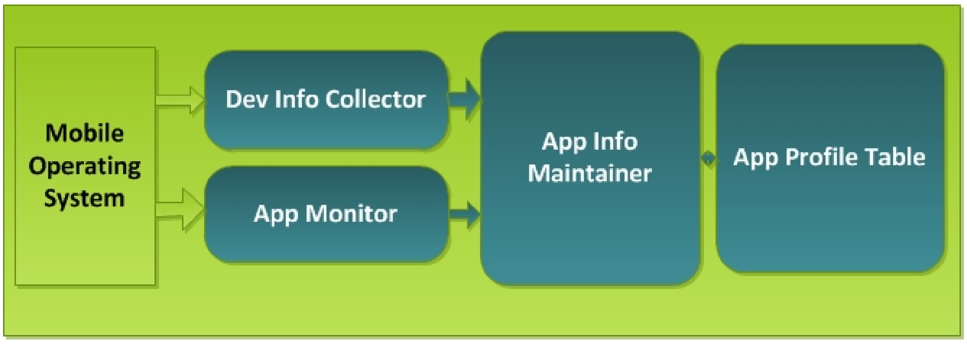

Figure 3.1: The Architecture of ASP

ASP uses different CPU energy models when CPU activates different number of cores.

the observed that the multi-core multi-frequency feature of modern mobile CPU can explain the non leaner relationship between CPU load and energy usage, which was not comprehended by existing researches [5].

3.3 ASP Architecture

This section presents architecture of the proposed ASP. Fig.3.1 illustrates the major components of ASP, and the information sharing mechanism among them for energy efficient routing for wireless communications. The highlighted components comprise the structure of ASP.

ASP works at application layer. *App Monitor* identifies the current running application each time a new application is launched. *Dev Info Collector* is responsible for fetching the status of the device from the *Mobile Operating System* by taking samples of multiple readings, including work load on CPU, load of wireless network card, cellular interface, characteristics of the display unit, and percentages of the battery remaining energy level. *App Profile Maintainer* combines the above information to create an energy-based application profile stored in the *App Profile Table*. Importantly, *App Profile Maintainer* keeps a timer in order to periodically update the profile records in the *App Profile Table*. In this manner, ASP provides energy constraints information for other mechanisms, such as AWERA, DEAS and EDCAM.

3.4 ASP Algorithm

3.4.1 Overview

Some existing energy modelling solutions are proposed generally for PC and smart phones. This does not suit the reality where mobile devices feature unique hardware structure and chip design as discussed in chapter 2. Even energy modelling techniques specifically designed for mobile devices do not consider tablet PCs for testing, yet tablets cover substantial market share of mobile devices as investigated by Cisco [3]. Moreover, they focus solely on hardware while overlooking the importance of user preference and software aspects. While users' habit affects the choice of application, the energy consumption of each hardware component shows distinctive features, and each typical application scenario (e.g. sending text message, watching video, etc.) shows distinctive energy requirements as well [8] [9]. Hence to keep record of applications power signature can ease the calculation of power constraint score from the energy model. This is because subsystem usage data can be collected less often as application type can dictate the typical power demand. Still the power signature needs update from time to time. The presented software solution addresses the above often overlooked issues.

Energy-oriented Application-based System Profiling (ASP) constructs a Profile that includes: *Energy Model* that takes workload of device components and calculates the corresponding battery discharge and *Application Profile Table* that records application power signature, namely the typical work load on the hardware components for each application.

First profiling makes use of *Component Workload Profile Table* to map between

”the work load on each hardware component” and ”the corresponding Power”. Next profiling technique is further developed with an online regression based Energy Model for accurate power estimation. Regression technique is a previously introduced technique for energy modelling [105][106][107]. In the proposed profiling process, it is used in both Initialization Phase and Working Phase.

The contributions of the proposed energy-oriented system profiling are: the application power signature based power estimation and multi-task scenario support. The use of application power signature enables an efficient energy constraint estimation without the expensive frequent power monitoring of the hardware. The energy model will produce the current energy constraint with little cost by knowing current application type and the corresponding power signature,

While the other Proposed energy modelling approaches [105][106][160] are not designed to deal with multi-task scenarios, our approach addresses this issue. The energy-oriented system Profiling in explained in detail next.

Table 3.1: Component Workload Profile Table

Workload	L_{CPU}	L_{GRA}	L_{CELL}	L_{WLAN}	L_{SCR}	P_{SYS}
5%	$L_{CPU}5$	$L_{GRA}5$	$L_{CELL}5$	$L_{WLAN}5$	$L_{SCR}5$	$P_{SYS}5$
15%	$L_{CPU}15$	$L_{GRA}15$	$L_{CELL}15$	$L_{WLAN}15$	$L_{SCR}15$	$P_{SYS}15$
20%	$L_{CPU}20$	$L_{GRA}20$	$L_{CELL}20$	$L_{WLAN}20$	$L_{SCR}20$	$P_{SYS}20$
...	...	...	...	...	...	...
90%	$L_{CPU}90$	$L_{GRA}90$	$L_{CELL}90$	$L_{WLAN}90$	$L_{SCR}90$	$P_{SYS}90$
95%	$L_{CPU}95$	$L_{GRA}95$	$L_{CELL}95$	$L_{WLAN}95$	$L_{SCR}95$	$P_{SYS}95$
100%	$L_{CPU}100$	$L_{GRA}100$	$L_{CELL}100$	$L_{WLAN}100$	$L_{SCR}100$	$P_{SYS}100$

The principle of system profiling was first introduced in the application-aware energy model of AWERA [24][161][162]. Then DEAS [148] develops the profiling techniques with Multi-task Support and used it for quality adaptation, which is an important element of this work.

3.4.2 Initialization Phase

Since devices of different model have different hardware specifications, the same application does not necessarily result in the same amount of workload on hardware components of different devices. Even the same percentage of workload does not necessarily show the same power readings on individual devices. Hence an initialisation phase is introduced to construct a *Component Workload Profile Table* prior to the construction of application profiles in order to make the proposed algorithm device independent.

In this phase, ASP runs a set of predefined tasks to put various loads on the different device components and monitors the corresponding power. For example, DEAS [148] runs a dedicated floating point addition program to load CPU to different degrees and monitors the corresponding power. Perrucci et al. [9] applied a similar approach to measure energy consumption of each hardware component of a smart phone with external circuitry. ASP takes the readings directly from the operating system and takes the value with better granularity. Although this phase introduces overhead, it is applied to each individual device once only. Besides, real tests show that online power monitoring and energy modelling can be realised with negligible overhead without affecting normal usage if designed carefully [106].

Among all the hardware components, the readings of the screen (SCR), the graphics processor (GRA), WLAN interface card, including WiFi (WLAN), cellular network interface module such as GSM for instance (CELL) and the processing chip set (CPU) are recorded only. This is because these components are the major energy consumers among the hardware components of the latest mobile devices (i.e smart phones or tablet PCs) and they show significantly higher energy consumption than the others[9].

The *Component Workload Profile Table* is shown in Table 3.1, where $L_{CPU}x$, $L_{GRA}x$, $L_{CELL}x$, $L_{WLAN}x$ and $L_{SCR}x$ (x = 5, 15, 20,..., 90, 95, 100) are workloads at x percent for CPU, graphics processor, cellular module, WLAN interface and screen, respectively. And $P_{SYS}x$ is the corresponding system power when the hardware components are at the workload of x percent. The workload of the system is represented as a vector in equation (3.1). ASP measures a small set of values to reduce the overhead.

$$\vec{L_{SYS}} = \begin{pmatrix} L_{CPU} \\ L_{GRA} \\ L_{WLAN} \\ L_{SCR} \end{pmatrix} \tag{3.1}$$

In equation (3.2), P_{sys} represents the utility function corresponding to the energy model of the current mobile system. The energy model is contributed by all the major device hardware components considered: CPU, screen, graphics, WLAN card and cellular module, respectively. L_{comp_i} represents the workload on the i-th device component. Weight values are used to balance the contribution of different hardware components on the overall utility function. Weight values W_{comp_i} are obtained by training the model with real workload and corresponding power values resulted from testing as shown in Table 3.1. c is a constant value used in multiple linear regression.

$$P_{sys} = \sum_{i=1}^{n}(W_{comp_i} \cdot L_{comp_i}) + c \tag{3.2}$$

The error function of the energy model is represented by equation (3.3). The goal of the training process is to minimise the calculation error ”E” of the energy model. This is realised by comparing the calculated power value P_{cal} and the real value from the OS P_{real} while adjusting the weight values in order to reach the optimal solution. P_{cal} is obtained from equation (3.2).

While the predefined loading tasks loading the system variously, the profiling procedure records a large set of observed workload combinations. Initially the training method calculates P_{cal} with one record from the observed data sets in conjunction with the initial weight values. Next many rounds of calculation take the record one by one from the observed data sets to calculate P_{cal}. In each round, weights values are redistributed for even smaller variation between P_{cal} and P_{real}. To avoid over-fitting, when error is lower than a threshold, the training stage finishes. The weight values are fixed for now to form an initial energy model for further energy constraint estimation. However this training stage takes place in "Working Phase" as well in order to adapt to the change of device usage.

$$Error = \frac{1}{2}\sum_{k}(P_{cal}(k) - P_{real}(k))^2 \tag{3.3}$$

In conclusion, the output of this phase is the energy model as in equation (3.2) with identified optimal weight values $\mathrm{W_{comp_i}}$. In the following phases, the energy model will take workload as input and output the estimated power level.

As explained above, the power consumption will be modeled as a function of the state of some predictors. There are several well-known methods which can be used, including non-linear methods such as Support Vector Regression (SVR) [163] and Neural Networks (NN) [163] and linear solution such as Multiple Linear Regression (MLR) [163]. The main advantage of using a linear method (MLR) is that it is easier to implement, however, the non-linear methods are more accurate (NN, SVR). In order to choose the best technique for our needs, the three regression models above were applied.

3.4.2.1 Regression Algorithms

Multiple Linear Regression

In our problem, the observed hardware component usage value x_i and the total power value of the system y_i are represented as paired data, e.g. $(x_1, y_1), \cdots, (x_n, y_n)$. The relationship between the observed predictors (hardware component usage values) and the total power value of the system is modeled as in equation (3.4), where ϵ is the random noise. The mean of ϵ is zero.

$$y_j = \beta_0 + \beta_1 x_j + \epsilon \tag{3.4}$$

ASP estimates β_0 and β_1 by minimizing the sum of the squared errors as presented in equation (3.5).

$$\sum_{j=1}^{n} (y_j - \beta_0 - \beta_1 x_j)^2 \tag{3.5}$$

The *predicted values* are obtained following iterations and they have improved accuracy with every iteration. During one such iteration, the predicted value is represented as $\hat{y}_j = \hat{\beta}_0 + \hat{\beta}_1 x_j$. The difference between the observation y_j and the predicted value $\hat{y}_j$ is called *error* as presented in equation (3.6). The mean square error (MSE) measures the performance of the prediction as in equation (3.7).

The open source Java library Weka[1] which implements the Maths tools that solve this problem and determine β_0 and β_1. The prototype system for real test is developed with WEKA using Java.

$$y_j - \hat{y}_j = y_j - \hat{\beta}_0 - \hat{\beta}_1 x_j \tag{3.6}$$

$$MSE = \sqrt{\frac{\sum_{j=1}^{n} (y_j - \bar{y})^2}{n}} \tag{3.7}$$

Neural Network for Regression

Fig.3.2 presents the architecture of the neural network regression approach. In our problem, the input nodes are the selected predictors which are the hardware compo-

[1] http://www.cs.waikato.ac.nz/ml/weka/

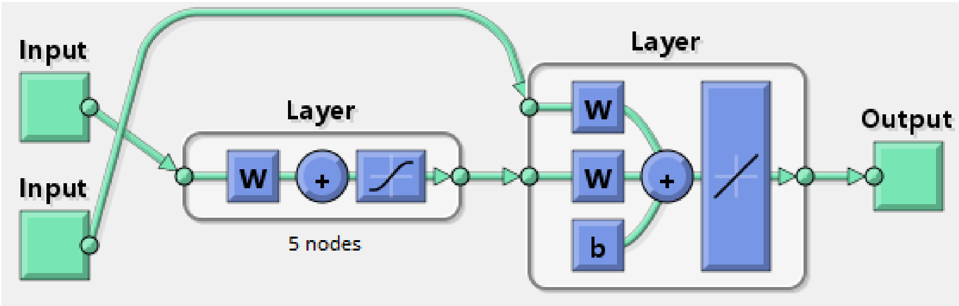

Figure 3.2: The Architecture of Neural Network for Regression

nent usage values represented in terms of percentage. At the first layer, the individual input values are assigned with different weight values w, and the weighted values are weighted again and summed in the last layer before output. The sum of the above weighted value is the output which is the estimated corresponding power of the system. This estimated power value will be compared with the actual value to calculate the error as in equation (3.6). The error will be used to adjust the weight values associated with each node. Eventually the weight values are fixed as coefficients for the resulted energy estimation model. In order to perform this computation, implementation uses Weka, an open source Java library [2].

Support Vector Regression

$$\bar{Y} = f(\bar{X}) = W \cdot \theta(\bar{X}) + b \tag{3.8}$$

$$\frac{1}{2} \parallel W \parallel^2 + C\frac{1}{N}\sum_{j=1}^{n}(L(Y_i, f(X_i)) \tag{3.9}$$

Support Vector Regression (SVR) is described in [164]. Equation (3.8) presents the relationship between the input predictors and the output estimation that SVR approximates. $\theta(\bar{X})$ is the high-dimensional feature space which is nonlinearly mapped from the input vector $\bar{X}$. $\bar{X}$ is a vector of the observed hardware component usage per-

[2] http://www.cs.waikato.ac.nz/ml/weka/

Table 3.2: Application Profile Table

	L_{CPU}	L_{DISP}	L_{CELL}	L_{WLAN}
App_j	$L_{CPU}(j)$	$L_{DISP}(j)$	$L_{CELL}(j)$	$L_{WLAN}(j)$

centage values which are the same as X_i in equation(3.4). A regularized risk function is introduced as in equation(3.9). Coefficients W and b are determined by minimising equation(3.9). In regularized risk function equation(3.9), the first term determines the coefficients and capacity of the function. The second term defines the penalty as follows: $L(Y_i, f(X_i))$ is 0 if $\parallel Y_i - f(X_i) \parallel \leq \varepsilon$, $L(Y_i, f(X_i))$ is $\parallel Y_i - f(X_i) \parallel -\varepsilon$ otherwise. This means the regularized risk function tolerates a range of predictions to be valid which means the predictions beyond the predefined range ε are considered invalid. The regression will use the valid predictions to calculate optimal coefficients. The implementation has employed Weka [3].

3.4.3 Monitoring Phase

Monitoring Phase constructs the *Application Profile Table*. Once a new application with no previous record is launched, ASP records the extra workload on the mentioned hardware components on top of the exiting figure caused by the applications on hold. Once the application shuts down, the average value is calculated and recorded as a new entry in the *Application Profile Table*. A streaming application delivering different data rates is regarded as a separate record, so that the energy constraint imposed by delivering different multimedia quality content is calculated separately. Each application is assigned a vector of its typical workloads as its power signature. The following description explains: how to use a vector to represent the power signature, and how this application profile is used in conjunction with the energy model described in the above section for energy constraint calculation.

Table 3.2 shows one simple implementation of the *Application Profile Table*. For

[3] http://www.cs.waikato.ac.nz/ml/weka/

application j, $L_{CPU}(j)$ gives the workload of CPU. $L_{GRA}(j)$ for graphic card, $L_{CELL}(j)$ for cellular interface and $L_{WLAN}(j)$ for WLAN card. Notably, the workload of the screen is not recorded as the brightness is highly dependent to the illumination of the environment and user preference. The above work load values are represented in a vector as in equation (3.10), where PS_{APP_j} is the power signature of application$_j$. Consequently, the value of power consumption for all running applications gives the power signature of the system. When energy constraint value is needed, the result of computations from equation (3.11) is the input of the energy model described in equation (3.2). The output of equation (3.2) is the running power of the whole system.

$$\vec{PS_{APP_j}} = \begin{pmatrix} L_{CPU}(j) \\ L_{GRA}(j) \\ L_{WLAN}(j) \\ L_{SCR}(j) \end{pmatrix} \tag{3.10}$$

$$\vec{PS_{SYS}} = \sum_{j=1}^{n} (\vec{PS_{APP_j}}) \tag{3.11}$$

Compared with constant hardware level monitoring, the proposed application-aware energy profiling is an easy and inexpensive approach as deployed devices will recognise the applications and use the power signature records in conjunction with the energy model to calculate the current application energy constraint. The application profile table is used to update model and signature.

3.4.4 Working Phase

This phase focuses on improving the existing application profile. Once a device encounters an application with a previous record in the *Application Profile Table*, it references the table for data to be used in order to calculate the current energy constraint.

In the working phase the *Application Profile Table* is updated incrementally. ASP occasionally monitors by sampling the average power on each component for applications which already have records in the *Application Profile Table*. According to (3.12), the updated workload $PS_{updated}$ for an application is calculated by the old value PS_{old} taken from the application profile and the new value PS_{new} that has just been measured. Weight values W_{old} and W_{new}, that distribute the effect of the two workload values, are determined by the running duration of the application. Each time an application is launched, new readings are used to update existing profile to enable adaptive self-learning.

$$\vec{PS_{updated}} = W_{old} \cdot \vec{PS_{old}} + W_{new} \cdot \vec{PS_{new}} \tag{3.12}$$

$$\frac{W_{old}}{W_{new}} = \frac{Duration_{old}}{Duration_{new}}$$

Importantly, the training process introduced in the Initialisation Phase is invoked periodically so that the energy model is able to perform accurately and adapt to the environment change of device usage. It results in overhead to collect system readings from mobile operating systems. Besides both the model training and energy constraint calculation result in overhead. Although the device periodically needs to obtain the actual power value from the system for model training, the overhead reduction is substantial given that energy constraint is often updated and used.

All the context information is learned and maintained via this self-learning process. The workload of each hardware components is monitored and sampled by the process for each application on current device, so that the application-related energy constraint is recorded as application profile. Moreover, each time an application is launched, new readings are used to update existing profile to enable adaptive self-learning. By taking this approach, users are able to deal with the enormous upcoming applications on the market.

3.5 Software Implementation

The implementation of this work was made in Java-Android. Some typical use cases are implemented and discussed. The GUI is shown in Fig.3.3 and consist of two areas: the left panel displays the real power and estimated power for three major hardware components including WiFi, CPU and screen, and the right panel presents the buttons to control the model training and result saving. The architecture of our implementation is depicted in Fig.3.4. It mainly is consists of five classes, four of which represents the statistics of CPU, Battery, WiFi and Screen. The main class implements all the functionality, including device monitoring, model training and device energy constraint estimation.

Figure 3.3: Graphical User Interface

I created a class for each subsystem analysed, and they are all controlled by the class *Amostrador*. To effectuate a new reading of each subsystem state, the *Amostrador* Class calls the methods *set* of each class, and after that the value is obtained with the method *get*.

The class *Amostrador* is also responsible for saving the values estimated and plot

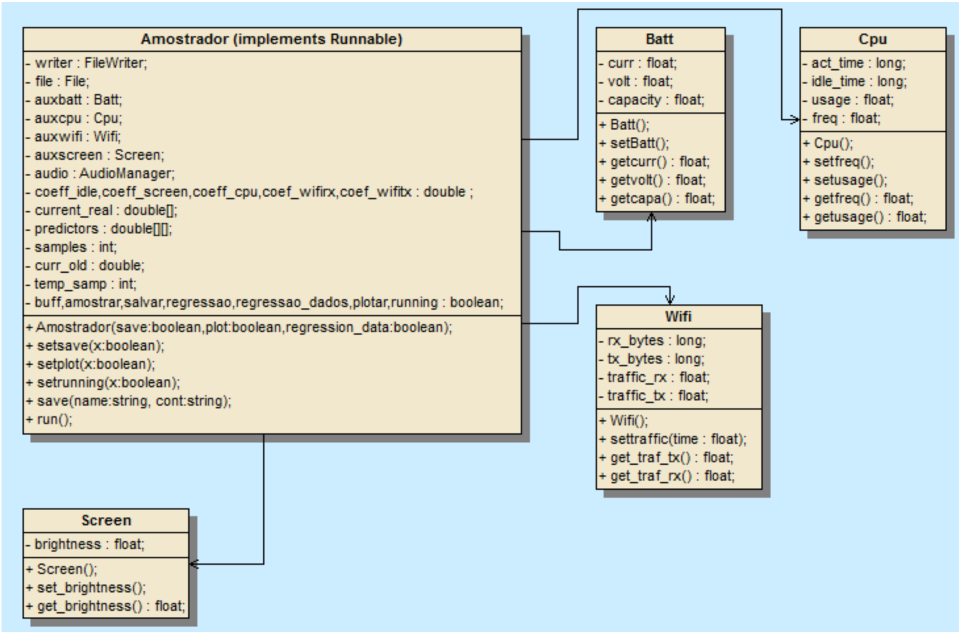

Figure 3.4: Class Diagram of Java Implementation

them into a graphic. *Amostrador* collects data for a new regression when it is running and a new regression can be performed when the user desires. The class *Amostrador* is controlled by an *activity* class with which the user can interact with a graphical interface.

3.6 Predictor Selection for Regression-based Modelling

3.6.1 Real Tests

In order to choose the best predictors to be taken into account for our model, different applications that control each component (while others are turned off) were run, and the battery current discharge was stored; such analysis is important in order to identify the best predictors and the relations that will be used. For the proof of concept purpose, four main subsystems of the tablet were considered: Screen, CPU, WiFi and Audio. Cellular and graphics will be included in the next release. These components were chosen since they show major impact on energy consumption during tests. To derive a more accurate model, other components such as bluetooth, GPS, 3G, could have been analysed. According to our tests, some components like Audio, they all have binary states, namely on and off. Once turned on, they all show nearly constant battery discharge, regardless of detailed changes. Therefore, Audio only is analysed in our work. The same methodology can be applied to other ”binary-state” modules, such as GPS.

Testing Scenario Description

Utility applications are developed to harvest the relevant data, including CPU frequency, CPU usage, WiFi packet rate, WiFi bit rate, battery current, battery voltage, audio state, screen brightness, etc from Samsung Galaxy Note 10.1 tablet. Dedicated applications are developed to load CPU and WiFi NIC to different extents. The above mentioned Android applications were developed on a laptop and installed on a Android smart phone and the Android tablet. A wireless router is in place for the real tests on WiFi NIC. In these tests, the Android smart phone and the Android tablet transmit data packets at different bit rates and packet rates between each other. An application is deployed on the tablet to collect the relevant results for inbound traffic and outbound traffic respectively. The whole set-up is shown in Fig.3.5. The USB cable connecting the laptop and the tablet is disconnected during tests so that the tablet is always powered by battery.

Screen

The screen power consumption model is derived using a training program that changes the brightness from 0 % to 100 % using 10 % step intervals each five minutes. The color of the background was also analysed, so the tests were performed for three different background colors (white, black and blue). As observed from the tests, the colour change results in very slight power changes only. Therefore, blue is used to represent

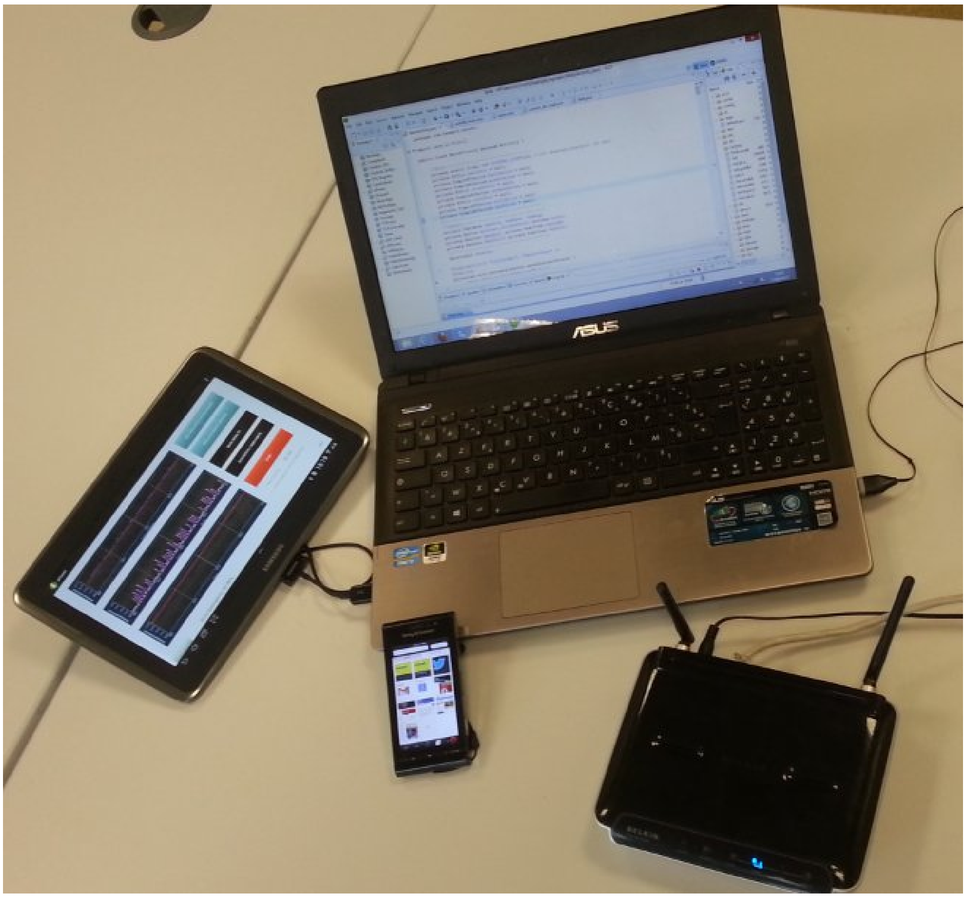

Figure 3.5: Real Tests : Set-up of Devices

the colour background. All the other components were turned off (WiFi card ,sound, Bluetooth) and the CPU was kept to a low load level, the battery current discharge was recorded.

The results can be seen in the Fig.3.6 below. It is noticed that the brightness is related in a linear way to the power consumed, however, changing the color of the display results in a slight variation of the power consumed, and although interesting, it will not be taken into account for the model since it is a slight variation and the tablet must be rooted to have information of the screen color, which can be an inconvenience for some users. The reason for such a behaviour is that LCD displays have a passive nature, most part of the energy used is to control an external lighting in contrast to OLED displays [165]. The color model for an OLED display can be done as proposed in [166]. The proposed energy model is shown in equation (3.13), where current discharge changes linear with the change of the brightness regardless of the detailed change of colour. The relation between screen brightness and battery discharge is shown in Fig.3.6: *Power* is linear to the percentage of brightness settings *Brightness*, and C_{Screen} is the coefficient.

$$Power = C_{Screen} \cdot Brightness \tag{3.13}$$

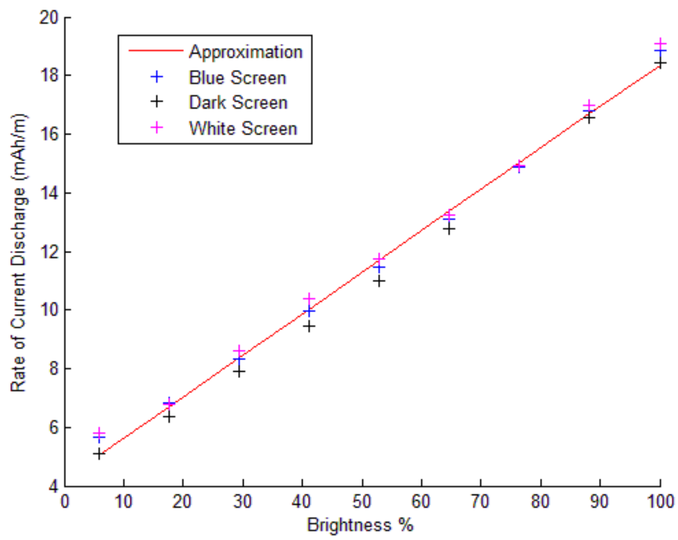

Figure 3.6: Relation between Battery Discharge and Screen Brightness

CPU

Many studies indicate that to estimate the power consumed by the CPU it is needed to analyse the load and the operating frequency of the CPU. In order to derive such a model for the CPU, an application written in Linux Script was used, this script executes an infinite loop and sleep during a certain amount of the period of this loop; changing this "sleep time", permits us to change the CPU load. This application was run, changing the CPU load at every 5 minutes, and the results were recorded. This test was repeated for 3 different CPU frequencies (200MHz, 800MHz and 1,4GHz), the CPU frequency can be set if the tablet is rooted. Some results are shown in Fig.3.7.

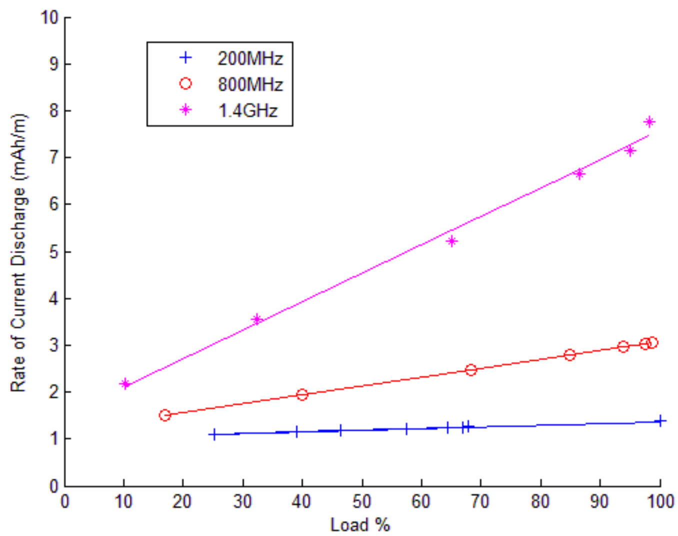

Figure 3.7: Relation between Battery Discharge and CPU Workload

It can be concluded that for a given frequency, the power consumption is linear with the load. This behaviour was also observed by [105], their solution to overcome this problem is to use two different predictors. In our case, the available frequencies are many from 200MHz up to 1400MHz. 100MHz steps are used; but considering a predictor for each frequency could lead to a dissolution of the data, so in order to derive a better model, more tests were performed and regression was conducted for each frequency. The model was derived as in Fig.3.8.

Next, two linear functions are considered to model the CPU. Frequencies from 200 MHz to 800MHz will be modelled as a ”LOW FREQ”, and frequencies from 900 MHz to 1,4 GHz will be modelled as ”HIGH FREQ”. The results are shown in Fig.3.9.

Finally, putting together all the results previously described, a strange behaviour noted when the CPU frequency is not previously set and is controlled by a a dedicated ”CPU frequency governor component” can be explained. If being configured in on demand mode, the governor will change CPU frequency on demand. Given the power

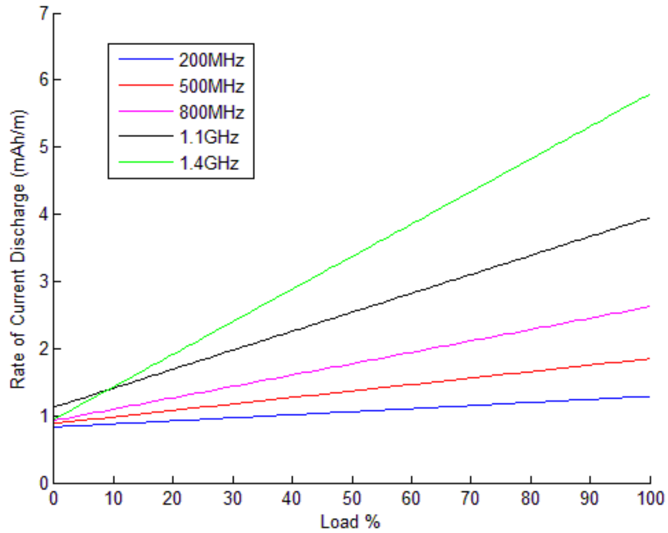

Figure 3.8: Relation between Battery Discharge and CPU Frequency

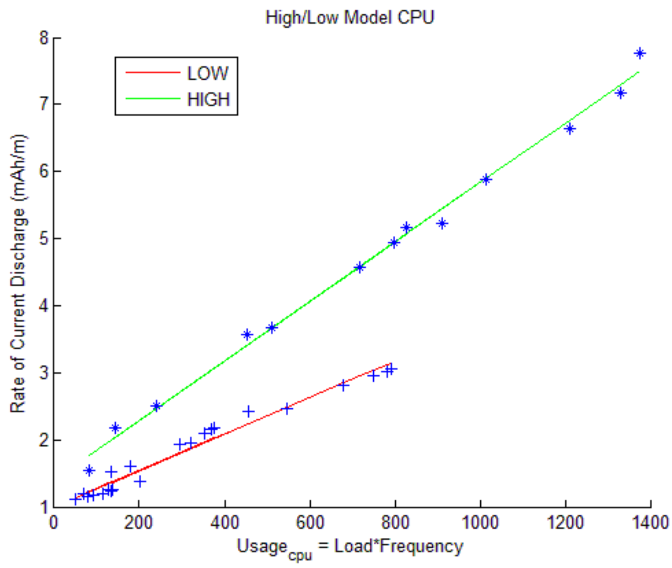

Figure 3.9: Relation between Battery Discharge and CPU Workload - Two-stage Frequency Setting

consumed in function of the load, a big leap of power for high values of load can be observed, this behaviour was also noted by [5]. However, observing the frequency, such a leap by a changing of the frequency is made by the dedicated ”CPU frequency governor component” as illustrated in Fig.3.10 and Fig.3.11.

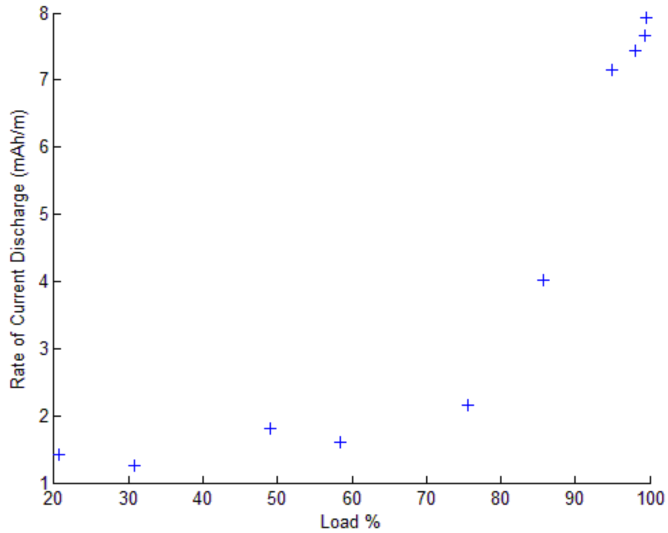

Figure 3.10: Relation between Battery Discharge and CPU Workload without Knowing Frequency Settings

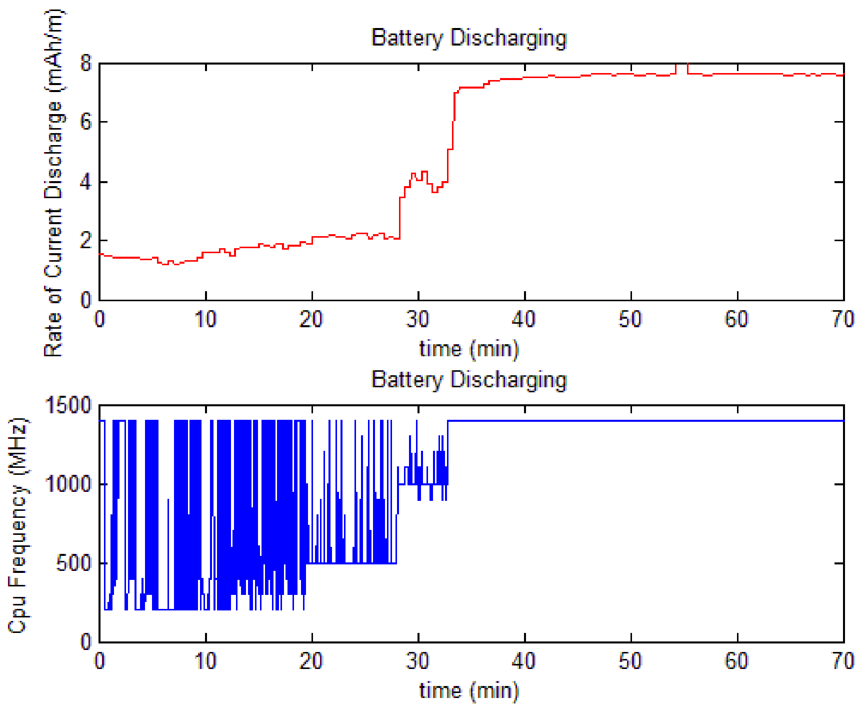

Figure 3.11: Relation between Battery Discharge and CPU Frequency Settings

WiFi

The transferred data between a WiFi connection causes an increase in the power consumed by the battery, however which indicator (bit rate or packet rate) should be used to predict its power consumed is a question that must be answered.

In order to find out the relation, to choose the predictor used, and to model the power consumed by the WiFi module, an application that permits the communication between the tablet and a computer via UDP packets in a LAN (Local Area Network), changing the rate of the transmission and the length of the packets (screen is kept off during the experiments) was built. Since the main reason of the present work is to create an energy-aware application to stream video (which is usually done by UDP packets), UDP packets were used instead of TCP. However, other works indicates that the results are similar for TCP packets [105].

To analyse the outbound traffic (from tablet to computer), the application changes its rate every 3 minutes (5 packets/s up to 120 packets/s) and the application was run three times with different packet lengths (614B, 1168B, 1568B). It is observed that the WiFi module activity and the CPU load are very related. Hence, in order to analyse the WiFi impact, the discharge variation caused by CPU is subtracted from the total power using the model previously described.

The results are shown in Fig.3.12, Fig.3.13 and Fig.3.14. It is clear that using the bit rate instead of the packet rate is a more accurate linear model and will lead to better results, therefore I use it for the inbound traffic. Other works suggest a state-based model [167], I believe that this kind of model could improve the accuracy in estimating the power, however, they are not as practical deploy-wise. This is due to the often limited refreshing rate of battery interface and the lack of knowledge of individual battery. Our effort aims at making the model generic and practical. The results are shown in Fig.3.12 and Fig.3.13. To analyse the inbound traffic (from computer to tablet), similar tests were done, an application changes its rate every 3 minutes (5 packets/s up to 120 packets/s) with a constant length. The results are shown in Fig.3.14. The WiFi energy model is demonstrated in equation (3.14).The *Power* is determined by both receiving data rate $bitrate_{rx}$ and transmitting data rate $bitrate_{tx}$.

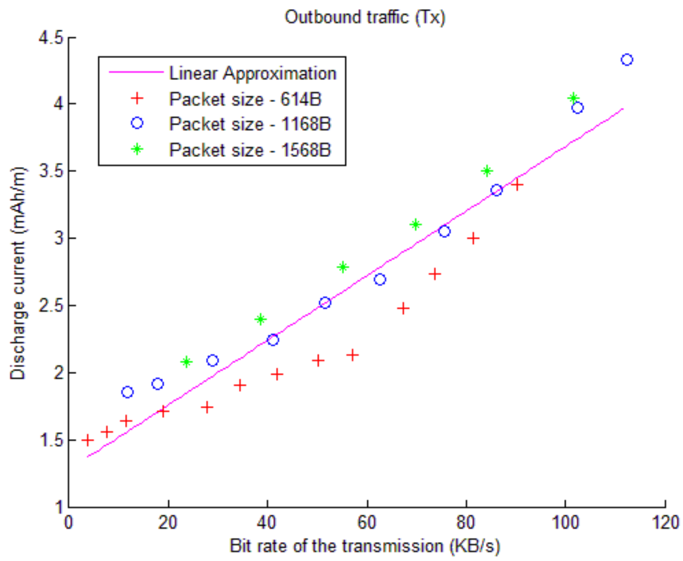

Figure 3.12: Relation between Battery Discharge and Bit Rate (Outbound Traffic)

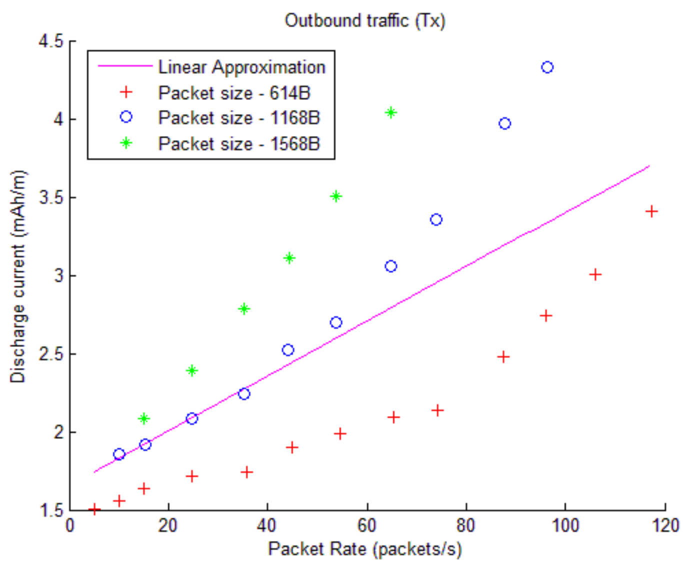

Figure 3.13: Relation between Battery Discharge and Packet Rate (Outbound Traffic)

$$Power = \beta_{\text{rx}} \cdot (bitrate_{rx}) + \beta_{\text{tx}} \cdot (bitrate_{tx}) \tag{3.14}$$

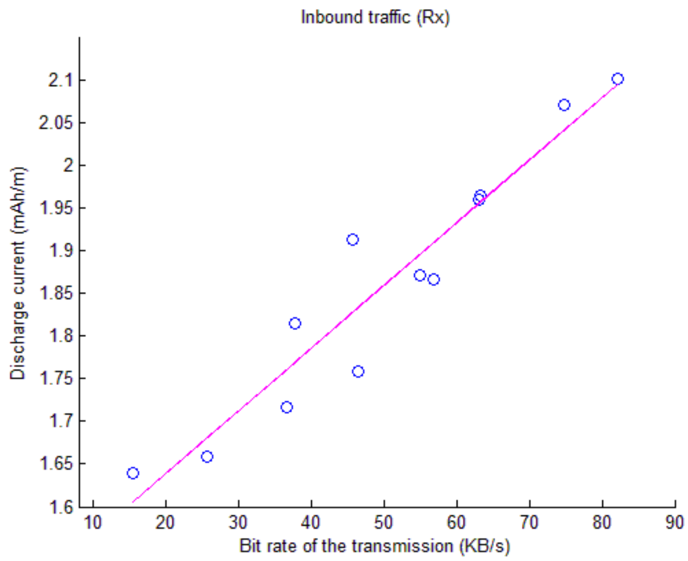

Figure 3.14: Relation between Battery Discharge and Bit Rate (Inbound Traffic)

Binary-state Module

Different audio volumes were set and a music was played during 8 minutes for each volume level, after that, the audio was turned off during 8 minutes. There is no big variation of the power consumed for different audio volumes, so the model considered is demonstrated by equation (3.15). In equation (3.15), $\text{State}_{\text{audio}}$ uses 1 and 0 to indicate on and off respectively. Other components such as Bluetooth, WiFi, GPS, network 3G can be analysed similarly.

$$Power = \beta_{audio} \cdot (State_{audio}) \tag{3.15}$$

3.6.2 Observations

The new observations from our field tests are presented. New approaches are proposed and implemented to address the issues overlooked by other researchers.

LED and OLED have different energy characteristics in terms of effect by colour changing. Murmuria et al. [166] state that colour affect the screen's energy depletion.

However, in our test conducted on our tablet equipped with LCD screen, only the change of brightness have clear affection. This observation has theoretical proof as in [165].

[5] discovered a strange behaviour regarding the relation between CPU frequency and battery discharge. The same effect is observed from our field test. This phenomenal is due to the frequency settings of modern mobile CPUs. CPU often has a dozen frequency settings that is controlled by CPU governor. CPU disables one or more cores when the CPU is not fully loaded in order to conserve energy. To address this issue, an accurate energy model for mobile CPU was proposed.

There have been discussions on the choice of predictors for WiFi module. Bit rate and packet rate are both used in the existing works. According to our observations, bit rate gives more accurate model than packet rate. This is because the packet size settings can vary for different applications. However the bit rate provides an unified quantitative measure for such scenarios. In addition, wireless NIC has two settings for WiFi transmission. A changing point of bit rate can be observed, beyond which NIC are set to high state that features high energy consumption and high throughput.

Voltage discharge curve [105] does not provide good indication of energy consumption here in the tests conducted on our Android tablet.

3.7 Regression-based Model Training

Once all the predictors and their relation to the consumed power are chosen, a regression must be performed for each different user, since the battery consumption varies among different uses. This regression can be re-done whenever the user desires, or when the error between the estimated power and the real power is too large for a certain time period; it means that our model is self-updated in order to achieve better results. Even if a different regression must be performed for different users, a training stage was performed in order to deliver a software that can be used in a initial time.

3.7.1 Training Strategy

Once all the predictors and their relation to the consumed power are chosen, a regression must be performed for each different user, since the battery consumption varies among different users. This regression can be re-done whenever the user desires, or when the error between the estimated power and the real power is too big for a certain time period [106], it means that our model is self-updated in order to achieve better results. Even if a different regression must be performed for different users, a training stage was performed in order to deliver the software with some coefficients pre-calculated that can be used initially.

The initial training strategy is summarised as following:

- CPU load from 100 % to 0 % (10 % steps), brightness from 0 % to 100 % (10 % steps) (1 hour) , over-crossing values can lead us to better results ;
- WiFi transmitting changing rate (same changes previously detailed), brightness constant (70 %) (1 hour) ;
- WiFi receiving changing rate (same changes previously detailed), brightness constant (50 %) (1 hour) ;
- Play a music with volume changing from 0 % to 100 % (20 % steps), brightness constant (60 %) (30 min) ;
- Idle state, screen is turned off, low use of WiFi, and the CPU is maintained in low frequency (1 hour).

After the initial training, the model is also trained using data collected from daily device usage. This mechanism allows adaptive model update.

Table 3.3: Error Rate of Models Using Different Target Value for Training

Training Target	SVR	MLR	NN
Power (mW)	0.62%	5.25%	4.52%
Current (mAh/min)	1.69%	3.01%	5.95%

3.8 Results of Model Training and Model Validation

To analyse the error of our power model, several tests were applied. To calculate the errors, the prototype takes the averaged value (every 90s) provided by the Battery Interface from the Linux Kernel averaged as the real power. According to Sesame [106], this result approximates the real value.

In the real tests, several aspects were evaluated: prediction using current value v.s. using power value; component utilisation based estimation v.s. power signature; the performance of ASP for video playback using video contents of different quality levels; overhead analysis.

3.8.1 Prediction Using Current Value v.s. Using Calculated Power Value

To begin with, the difference of using current (mAh/min) or the real power (mW) as the target of model training was analysed. A test using the application YouTube was performed and results can be seen here below. Next are some observations: a) Support Vector Regression (SVR) is the better method for both tests, however it was more accurate for power prediction instead of current prediction. b) Multiple Linear Regression (MLR) had better results for the current prediction. c) Neural Network (NN) gave us the worst result in general, but as is shown, the results from this method vary a lot with different trainings (with same training data); for our purposes, this method is worse than the others.

Fig.3.16 shows the error rate of models obtained from using different training methods (SVR, NN, MLR) with calculated power as training target. Fig.3.15 shows the error rate of models obtained from using different training methods with current

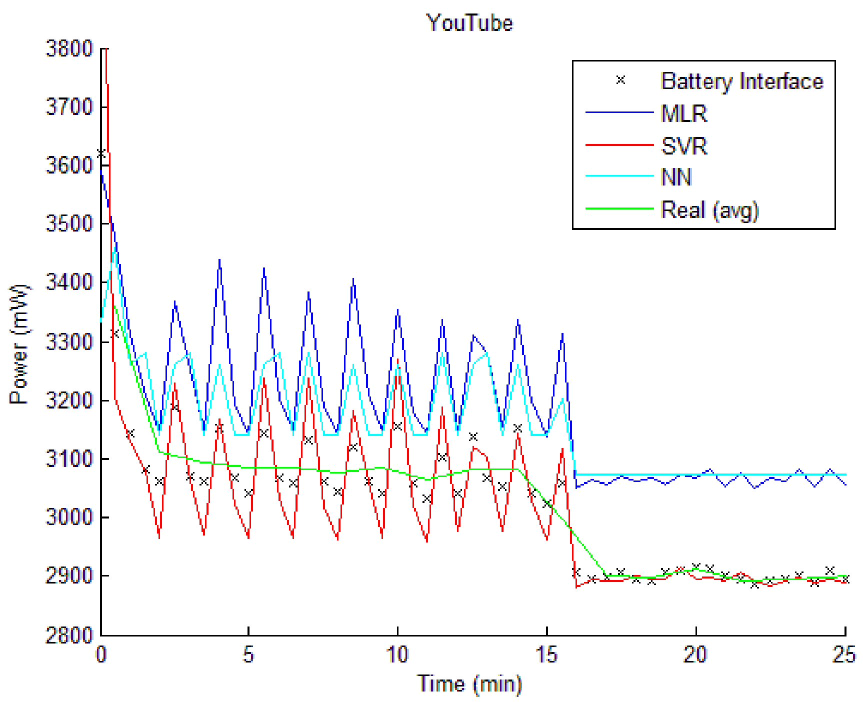

Figure 3.15: Estimations of Models Using Power as Training Target

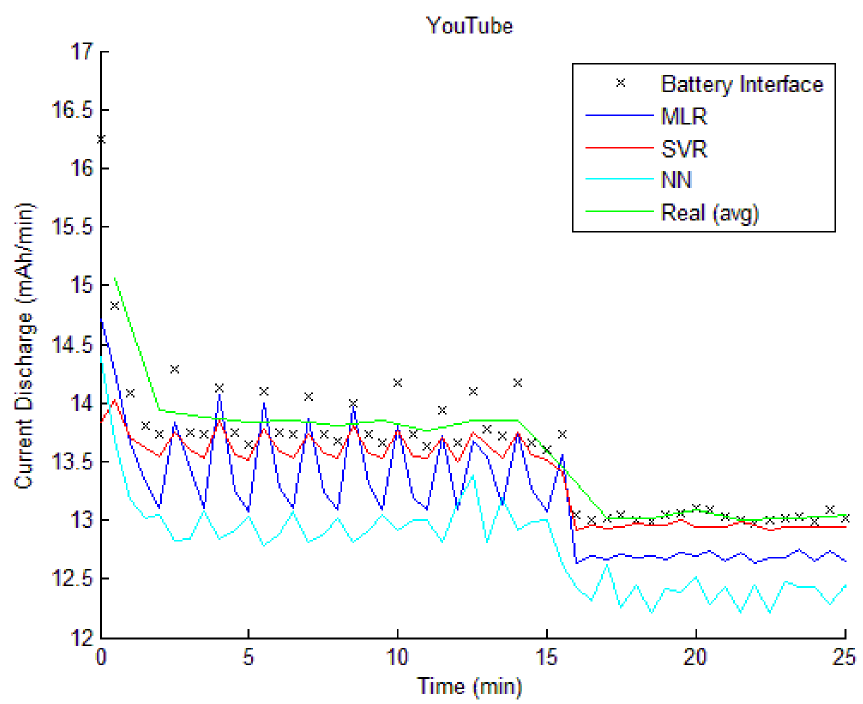

Figure 3.16: Estimations of Models Using Current as Training Target

Table 3.4: Error Rate of Models Using Different Training Applications

Training Applications	SVR	MLR	NN
Video Player and Angry Birds	0.62%	5.25%	4.52%
Predefined Tasks	8.89%	6.59%	13.31%

value as training target. Table 3.3 presents the exact results from the above test. Table 3.3 clearly shows SVR delivered most accurate results for both training strategies. Additionally using power value as training target demonstrates more accurate results for SVR and NN.

3.8.2 Training Using Predefined Tasks v.s. Incremental Adaptive Training Using Daily Used Applications

ASP uses predefined training tasks to load the system variously in order to obtain an initial model. The energy model of ASP incrementally trains itself when the device is put into daily usage. This enables the model to adaptively adjust itself to the specific user patterns in terms of choice of applications. Fig.3.17 presents the error rate values of models trained using predefined tasks. Fig.3.18 presents the error rate values of adaptively trained models. Table 3.4 presents the results from both figures, which clearly suggests the adaptive model constantly outperformed the model trained using predefined tasks. This demonstrates that it is necessary to introduce incremental adaptive training into user's daily device use.

3.8.3 Model Validation with Video Delivery

Since ASP is used by DEAS and EDCAM for adaptive video delivery, this section evaluates the energy model of ASP by a) playing two different video contents (Interview[4] and Surfing[5]) from YouTube with two different quality levels. b) performing local video (Top Gear 720p 60min) playback in Android VideoPlayer at three different

[4]http://www.youtube.com/watch?v=UPAxSGBPPOE
[5]http://www.youtube.com/watch?v=mreJwO2GQWg

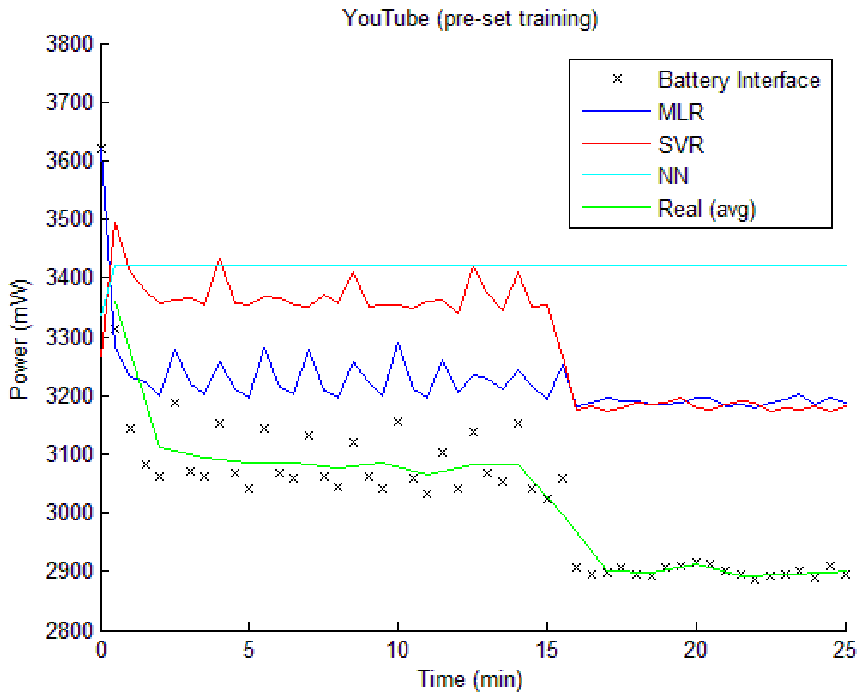

Figure 3.17: Estimations of Models Trained Using Predefined Tasks

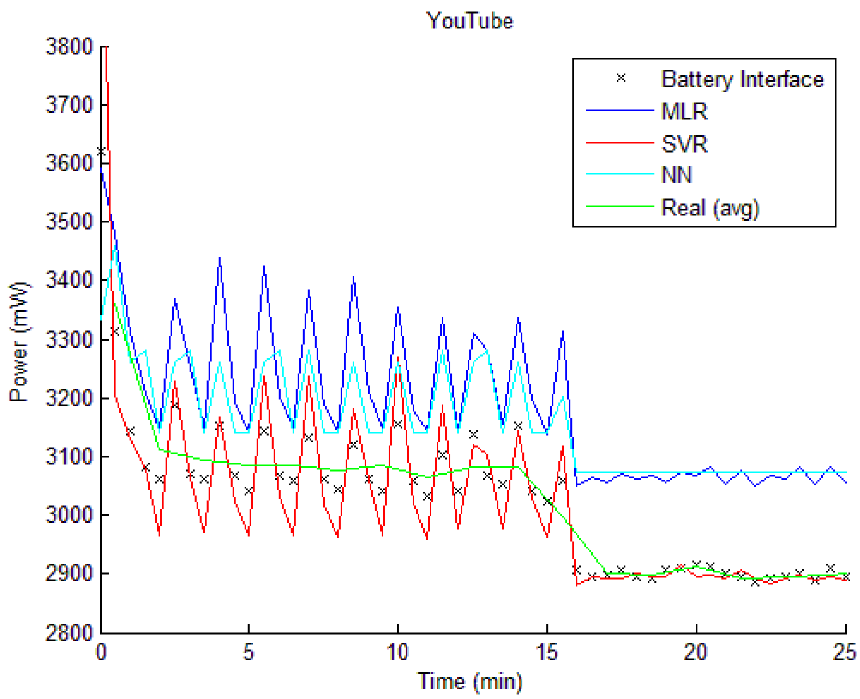

Figure 3.18: Estimations of Models Trained Using Daily Used Applications

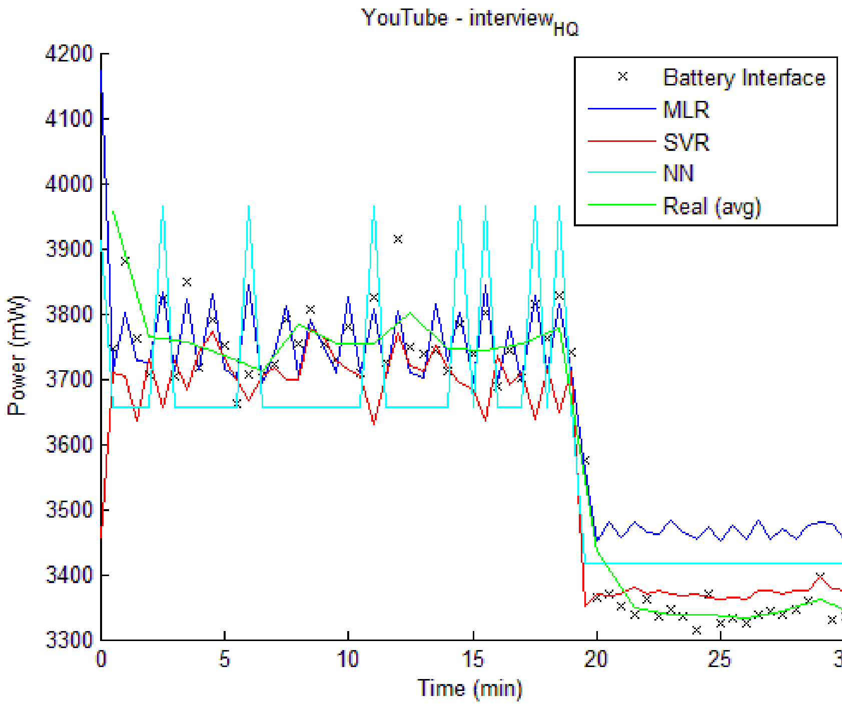

Figure 3.19: Estimations of Playing High Quality ”Interview”

Table 3.5: Error Rate of Playing ”Interview” At Different Quality Levels

Video Quality Level	SVR	MLR	NN
High	1.3%	2.1%	1.9%
Low	0.9%	2.3%	1.7%

quality levels.

Fig.3.19 and Fig.3.20 present the power estimations made by ASP while playing ”Interview” from YouTube at high and low quality levels, respectively. Fig.3.21 and Fig.3.22 present the power estimations made by ASP while playing ”Surfing” from YouTube at high and low quality levels, respectively. Fig.3.23, Fig.3.24 and Fig.3.25 present the power estimations made by ASP while playing ”Top Gear” from SD card

Table 3.6: Error Rate of Playing ”Surfing” At Different Quality Levels

Video Quality Level	SVR	MLR	NN
High	1.4%	1.7%	2.4 %
Low	0.9%	2.3%	2.1%

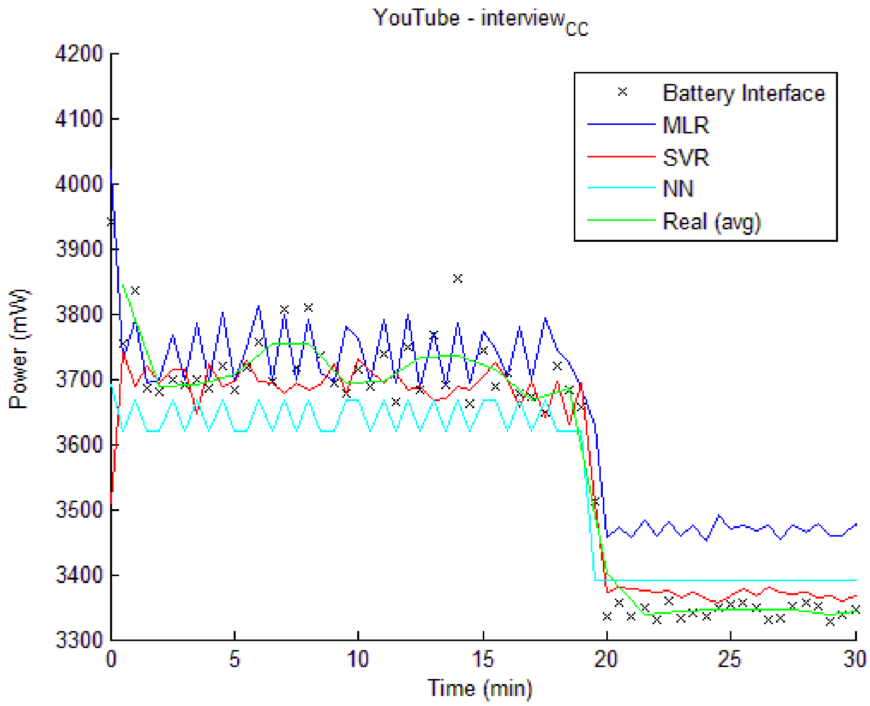

Figure 3.20: Estimations of Playing Low Quality ”Interview”

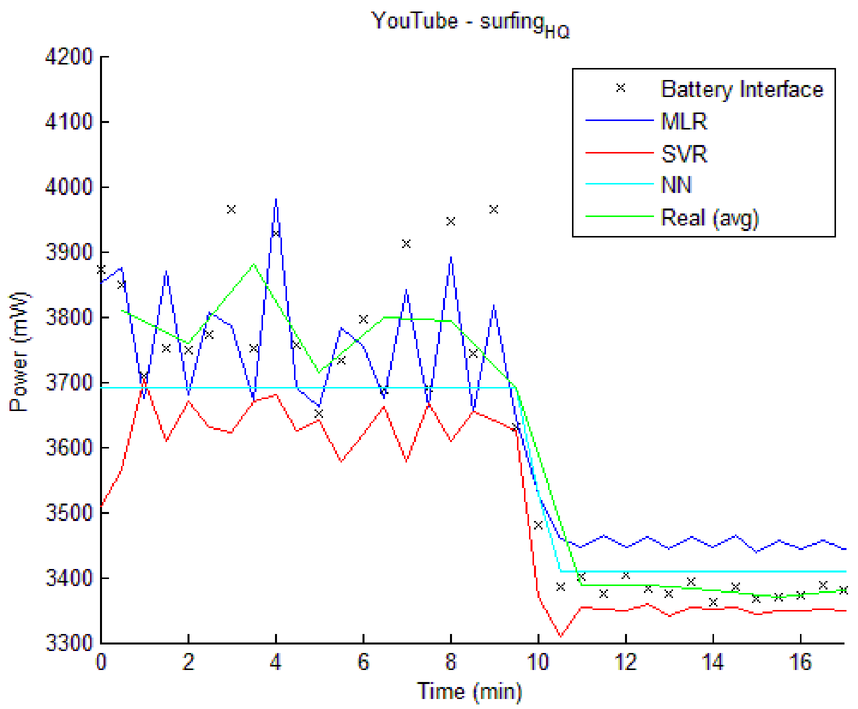

Figure 3.21: Estimations of Playing High Quality ”Surfing”

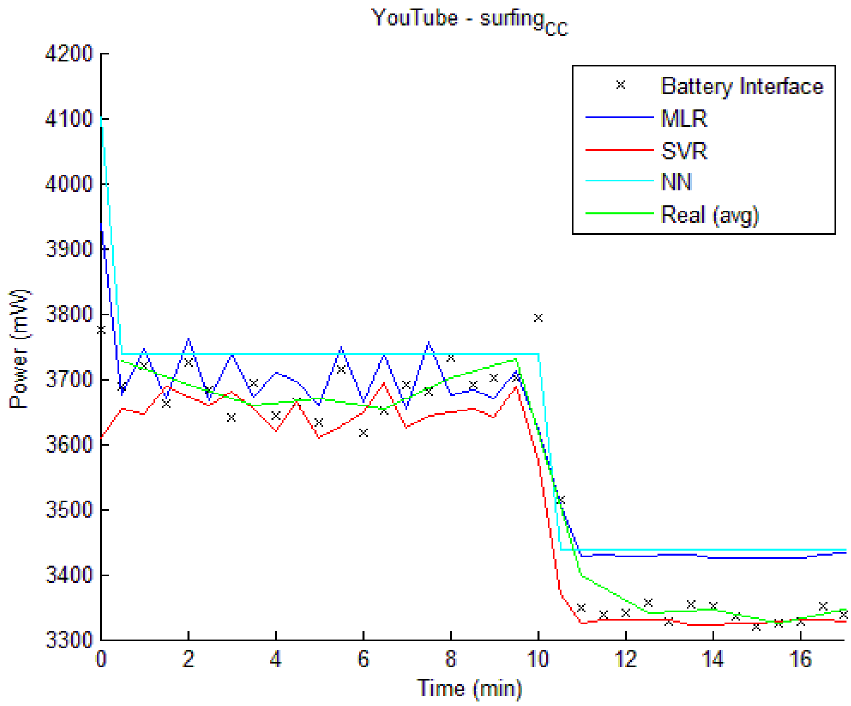

Figure 3.22: Estimations of Playing Low Quality "Surfing"

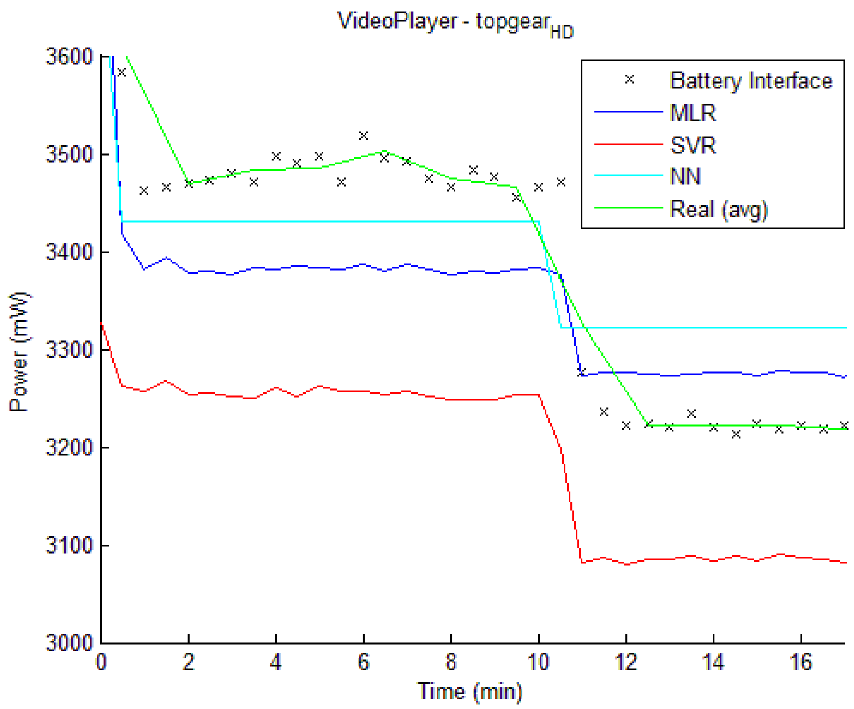

Figure 3.23: Estimations of Playing High Quality "Top Gear"

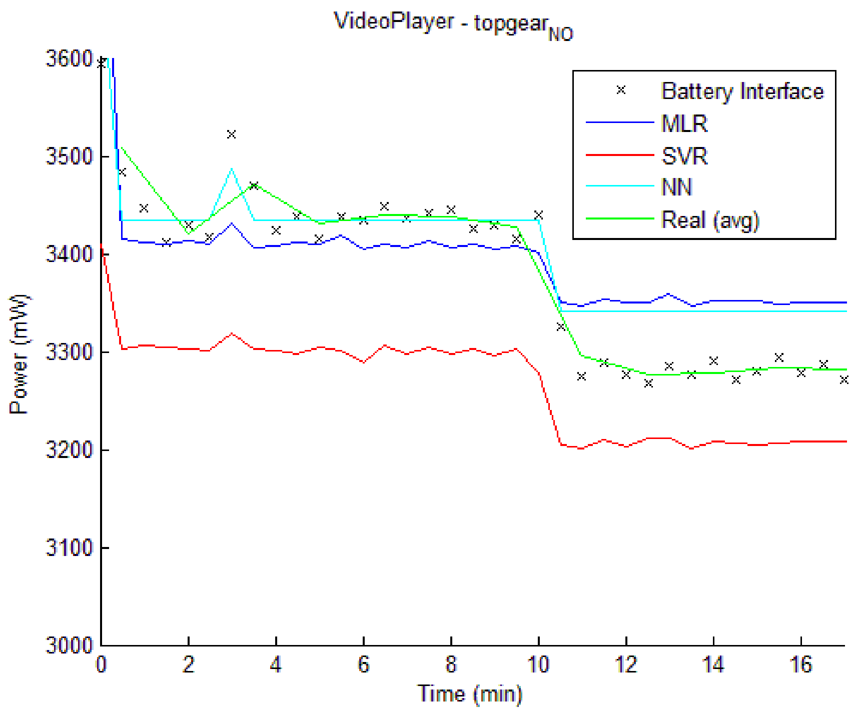

Figure 3.24: Estimations of Playing Medium Quality ”Top Gear”

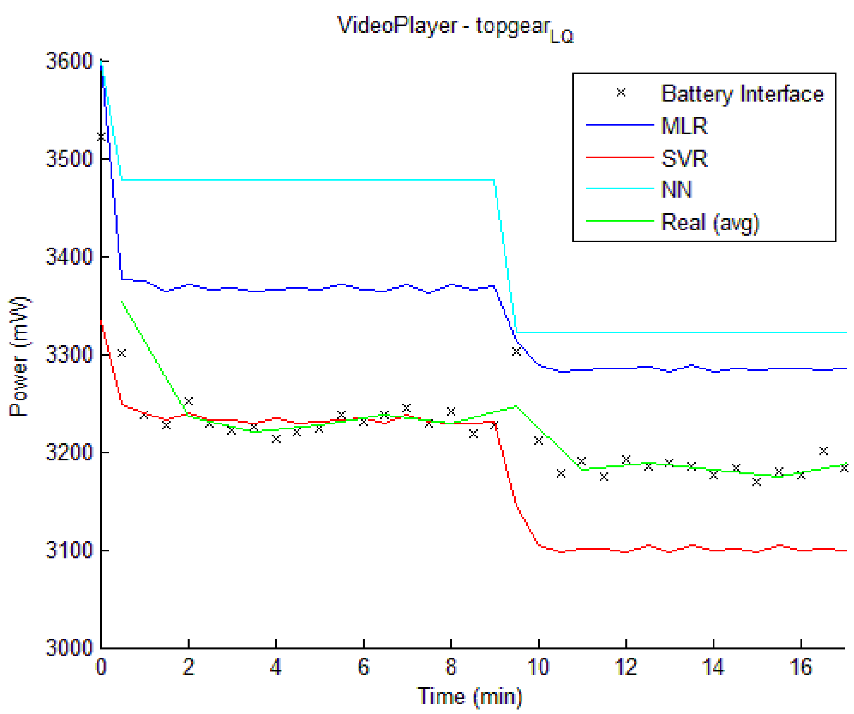

Figure 3.25: Estimations of Playing Low Quality ”Top Gear”

Table 3.7: Error Rate of Playing ”Top Gear” At Different Quality Levels

Video Quality Level	SVR	MLR	NN
High	2.1%	2.1%	5.7 %
Medium	1.4%	0.9%	3.3%
Low	1.7%	5.6%	3.4%

at high, medium and low quality levels, respectively. Table 3.5, Table 3.6 and Table 3.7 presents the exact error rate for the above testing scenarios. According to these testing results, ASP with SVR achieved the error rate below 3 percent. Both ASP with SVR and ASP with MLR demonstrates stable and constant results. In conclusion, SVR features lower total average error rate.

3.8.4 Utilisation-based Estimation v.s. Power Signature-based Estimation

Existing energy modelling techniques use utilisation-based estimation as mentioned in *ASP* chapter. However, ASP keeps record of power signature of individual application and makes energy estimations based on the power signature. Both methods are tested using the developed prototype. In this test, the power signature of YouTube from previous tests is used for power signature-based estimation. For utilisation-based estimation, the monitored real time utilisation values are used. For the test, the tablet played two clips ”Interview” and ”Surfing” from YouTube. Fig.3.26 presents the estimations by both methods. As presented in Table 3.8, both methods show low error rate of below 2.5 percent. Power signature-based method is used to reduce overhead and predict the power consumed. And it demonstrates lower error rate as well.

Table 3.8: Error Rate of Utilisation-based Estimation and Power Signature-based Estimation

Estimation Method	Error Rate
Utilisation-based SVR	2.5%
Power Signature-based SVR	2.2%

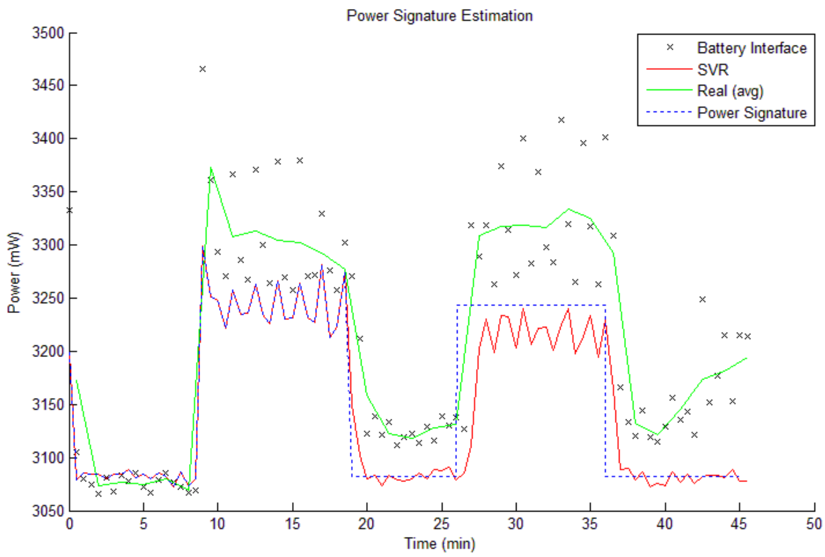

Figure 3.26: Utilisation-based Estimations (SVR) and Power Signature-based Estimations

Table 3.9: Overhead Results

Sampling Rate of Utilisation Values	Overhead
20 Hz	34.2 mW
10 Hz	10.2 mW
2 Hz	3.6 mW

3.8.5 Overhead Analysis

The overhead of ASP while it is configured to use three different sampling rates (20 Hz, 10 Hz, 2 Hz) to collect hardware utilisation readings was analysed. The overhead increases with the rate. Considering the overhead of Sesame is above 80 mW, the overhead of ASP is relatively low. Additionally, to apply power signature-based estimation will reduce the needed sampling rates. Consequently, the overhead can be further reduced. Table 3.9 presents the overhead values obtained from real tests on the prototype.

3.9 Chapter Summary

ASP includes three-phase Energy-oriented Application-based System Profiling for mobile devices which (a) constructs a mapping between the work load on each hardware components and the corresponding system power value; (b) monitors the work load on the hardware components for each application, in order to construct the application profile table; (c) calculates current energy constraint of current context by referencing the above tables and monitoring current status of the device (screen brightness, application type, etc). This chapter starts with an introduction of the proposed energy modelling technique. Compared with other proposal, the advantages and contributions of the proposed solution are introduced. Next, the details of the conducted real tests for predictor selection are presented, including device set-up and results from each hardware subsystem using different approaches. A brief summary of the tests results outlines the concrete data and reasons backing up the design of the proposed energy modelling technique. Following predictor selection, the regression algorithm, training strategy along with software implementation of the prototype system are explained in details. Last, a series of comprehensive tests for model training and model validation are presented.

Chapter 4

Application-aware Energy-efficient Routing Architecture, Algorithms and Testing

4.1 Overview

The latest mobile devices such as smart phones and tablet PCs, equipped with high resolution interactive screens, highly advanced CPUs, wireless networking and multimedia processing capabilities have become very important in people's daily life. The growth of these devices' popularity has determined an increased interest from shopping malls, theme parks, institutions, convention centres, etc. to deploy wireless network infrastructures and offer diverse online services addressed to such device users. However, infrastructure deployment is expensive and highly localized, so accessing content from mobile devices supported by ad-hoc wireless connectivity is considered a very good alternative solution. In such scenarios, energy efficiency has always been a key issue and is highly important especially for wireless routing algorithm designs as the mobile wireless devices are powered by batteries with limited power capacity. Moreover, different applications, e.g. online games, online chat, video streaming, etc. put various loads on different hardware components (e.g. CPU, wireless card, screen) and result in different energy constraints. Such energy constraints of different devices are measured and recorded by the innovative **ASP**, which enables functionality of device-differentiated energy efficient solutions as described in chapter 3. In this context, this thesis proposes a novel **Application-aWare Energy efficient Routing Algorithm** (AWERA) for heterogeneous wireless networks which performs energy-aware routing based on application-related characteristics and nodes' energy budget. AWERA makes use of ASP to keep track of nodes' energy depletion according to the application type they run, network load and remaining battery energy level. Based on ASP, AWERA performs energy-aware routing that dynamically makes route selection to improve network energy efficiency.

The context of smart device usage in wireless networks includes: application properties, device features (e.g. screen size, battery capacity, etc), network conditions and user preferences. This context is often energy related.

For example, different applications put different work load on the hardware and this results in different energy consumption. Compared with devices with smaller screens and larger battery size, those equipped with larger screens and smaller battery capacity suffer from shorter lifespan between recharges. Network condition is also a very important aspect of the mobile device usage. A wireless link with bad signal reception may need multiple re-transmissions before successful communication, which is energy consuming.

The device is aware about all the above-mentioned context information that can be accessed from the operating system at the application layer. As to the applications and device specifications, smart phones share a similar structure and the energy consumption of each hardware component shows distinctive features, and each typical application scenario (e.g. sending text message, watching video, etc.) shows distinctive energy requirements as well [8] [9].

Then this chapter presents the simulation-based testing environment and simulation scenarios used to fully evaluate the performance of AWERA. For each scenario, the simulation settings and scenario description are described. The schemes used for performance comparison are introduced as well. The metrics used for assessment during the testing are presented along with the testing results.

Compared with traditional wireless routing algorithms, AWERA introduces extra overhead. It needs periodical update to make sure it can make decision according to the latest energy distribution in the networks. However as far as I am aware of, the existing similar energy efficient routing solutions need periodical update process to ensure its energy saving policy. In addition, AWERA may face throughput reduction by choosing energy efficient routes. Consequently AWERA takes number of hops and link quality into consideration to address such issue.

The overhead for one node is computed as AWERA Routing Control Packet Size * Periodical Update Rate. The Control Packet Size is 24 bytes in this case. The Periodical update rate is configurable. Besides, the use of more energy efficient routes may result in degradation in terms of throughput as demonstrated in the section ”Simulation Testing and Result Analysis”. Since AWERA uses ASP to obtain device energy constraint information, the overhead of ASP is also included in the overhead of AWERA.

4.2 AWERA Use Case Scenario

A typical scenario for AWERA is depicted in Fig.4.1. In a very large public area, such as amusement park, shopping plaza, etc., it is difficult to maintain and manage high quality WiFi-based wireless networks [2]. This is due to the sheer size of the area to cover and high density of mobile users. Currently, 3G and 4G networks are the only available networking infrastructure in such scenarios.

However, some wireless network service users may not necessarily wants to use

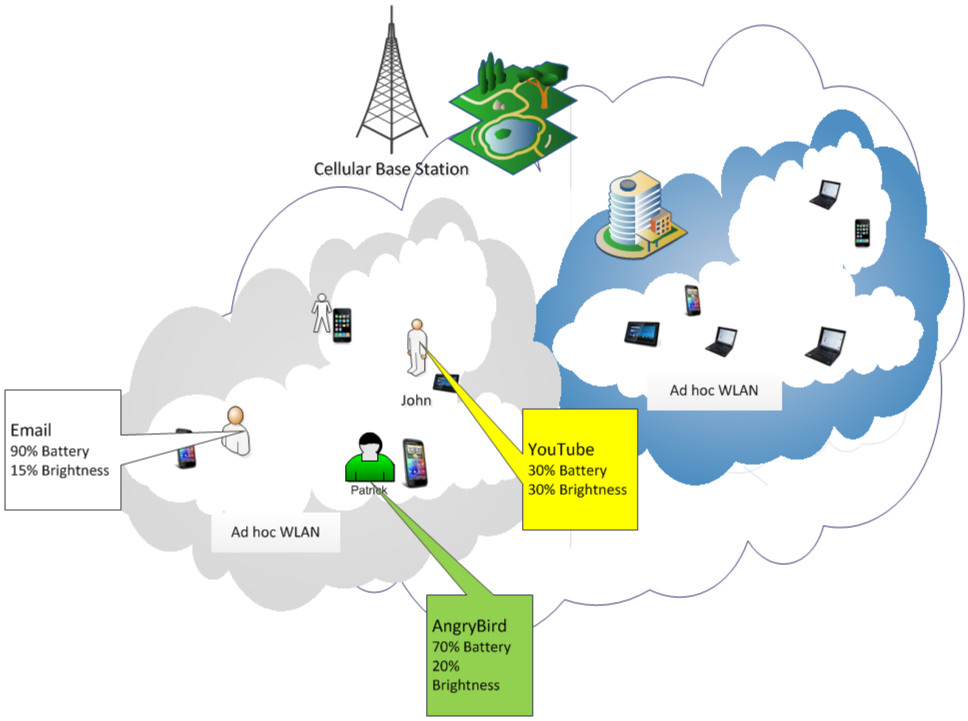

Figure 4.1: Example of Use Case Scenario for AWERA - Theme Park

the often very expensive cellular networks. For example, a group of tourists queuing up for a roller coaster ride can well form an ad-hoc WLAN using their already available wireless NICs of their mobile devices. In this context, ad-hoc WLAN is an inexpensive and flexible choice for them to play WLAN-based games, share information and pass some time in a funny manner. Another possibility is to set up an AP for local Intranet services to enable access to maps and tourism information for example. The AP can help organise all the user devices to form an ad-hoc WLAN, which has better coverage and suffers less congestion than the traditional AP-based WLAN. This theory has been accepted as useful in many emergency recovery scenarios, tsunamis, earthquakes, etc in which ad-hoc WLANs can play critical roles.

In Fig.4.1, the cellular base station provides user devices with the ability to access the Internet. However, no additional infrastructure-based WLAN is organised. Instead, people near each amusement equipment have organised their own ad-hoc WLANs. AWERA is deployed among these devices aiming at prolonging device battery life by using energy efficient ad-hoc routing. Moreover, AWERA tries to achieve load balancing to enable fair energy consumption among nodes.

To achieve the above objectives, the energy consumption in the context of device usage is recorded and referenced for route selection. The context of device usage includes application type, remaining energy level, screen brightness and so on. ASP creates and automatically maintains the energy-oriented application-aware device energy model. Using this model, AWERA considers the current energy constraint and performs energy-efficient sensible route selection.

Similar to all the other users near the roller coaster, Patrick's tablet keeps a record of its energy constraints. All the devices within the self-organized network will exchange their energy constraint on demand when route construction is needed. If all the devices nearby will relay Patrick's data to the final destination, it is highly likely Patrick's tablet will not choose John's device to forward the data. This is because John's device is running more energy consuming applications and uses brighter screen, resulting in lower remaining energy and tight energy budget. Instead Bob's device will be used which has higher energy budgets. This energy-aware routing is made possible by both ASP and AWERA.

4.3 AWERA Architecture

The architecture of the proposed AWERA routing solution is presented in this section. AWERA is a cross-layer context-aware routing protocol involving application and network layers. Fig.4.2 illustrates the major components of AWERA and ASP, and the information sharing mechanism among them for energy efficient routing for wireless communications. The highlighted components comprise the structure of AWERA.

Since routing decisions are made in the network layer, all the context related information is aggregated by the *Routing Info Centre* located in the Network Layer. *ASP*

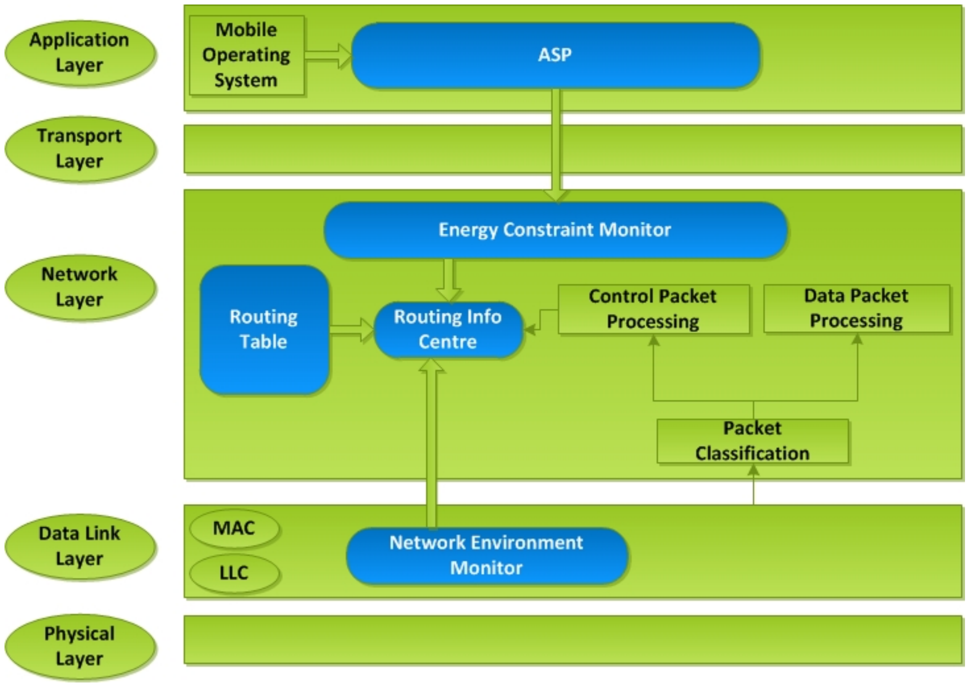

Figure 4.2: The Architecture of ASP and AWERA

notifies the *Energy Constraint Monitor* every time the Operating System finishes an old application and launches a new application. If the *Energy Constraint Monitor* can find the application profile for the current application, it will fetch that profile and passes it to the *Routing Info Centre*. Otherwise, it asks the *App Profile Maintainer* to create a new application profile. Meanwhile, the *Network Environment Monitor* monitors the link quality of the available links to neighbor nodes by detecting the signal strength of the incoming traffic. The weaker the received signal is, the more effort and energy are required for successful transmission to a neighbor node. Except for application layer feedback, link quality also needs to be evaluated for efficient routing decision making. This piece of link quality information is passed to the *Routing Info Centre* by the *Network Environment Monitor*. After the above process, *Routing Info Centre* will process all the context information and calculate the context-based energy constraint of current device and each available link. As a result, it maintains a *Context-aware Cross-layer Energy-efficient Routing Table*. The process of context monitoring and information processing is described in the next section.

The *Packet Classifier* module inspects all the incoming traffic, and hands over data packets to the *Data Packet Processing* module, control packets to the *Control Packet Processing* module. *Data Packet Processing* module will pass packets designated to the current device on to Transport Layer for local processing. *Data Packet Processing* module will check the *Routing Table* for the rest of data packets, and forwards the packet to the next hop if any routing record is available in the *Routing Table*. The *Control Packet Processing* module is a very important element of this routing protocol. It passes valid routing control packets to the *Routing Info Centre*. The *Routing Info Centre* will add fresher routes contained in the control packet to the *Routing Table*. Then the *Routing Info Centre* updates the control packet by adding the context-aware energy constraint of the current device to the routing cost field, and forwards the control packet for route construction.

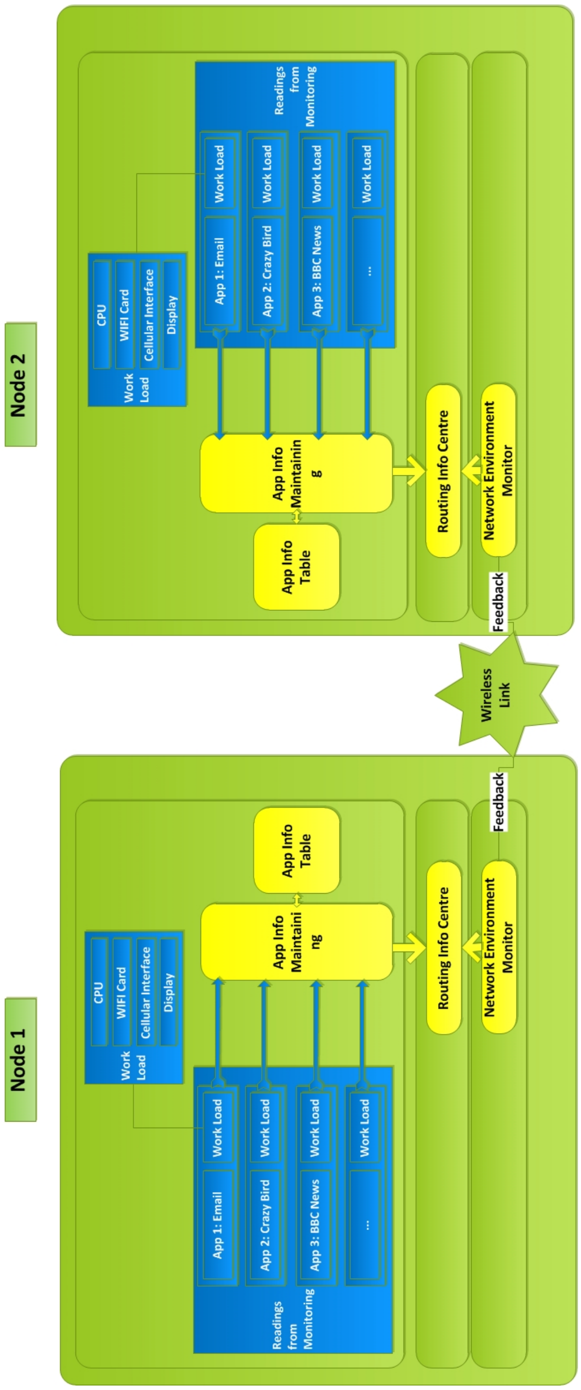

Figure 4.3: Cross-layer Mechanism of AWERA Working with ASP

Table 4.1: Calculation of Individual Components' Utility Functions

	G_{CPU}	G_{GRA}	G_{CELL}	G_{WLAN}
App_{j}	$\frac{E_{\text{CPU}}(j)}{MaxE_{\text{CPU}}}$	$\frac{E_{\text{GRA}}(j)}{MaxE_{\text{GRA}}}$	$\frac{E_{\text{CELL}}(j)}{MaxE_{\text{CELL}}}$	$\frac{E_{\text{WLAN}}(j)}{MaxE_{\text{WLAN}}}$

Table 4.2: Real Readings in Idle State when GSM and WiFi Are Used

	G_{CPU}	G_{GRA}	G_{CELL}	G_{WLAN}	G_{SCR}
Idle State	7%	80%	9%	1%	variable

Fig.4.3 clearly illustrates the mechanism of ASP and how AWERA cooperates with ASP. At Application layer, the mapping between application type and typical observed work load on each major hardware components are recorded by *App Info Maintaining* module, which stores the processed information in *App Info Table*. Meanwhile, *Network Environment Monitor* performs MAC layer feed back observing to evaluate the link quality of the available wireless link. The information collected from application layer and MAC layer is aggregated in network layer at *Routing Info Centre*.

4.4 AWERA Algorithm

4.4.1 Node Cost Computation Based on Application Profile

$$G_{app} = \frac{\sum_{i=1}^{n}(W_{comp_i} \cdot G_{comp_i})}{\sum_{i=1}^{n} W_{comp_i}} \tag{4.1}$$

$$G_{comp_i} = \frac{E_{comp_i}}{MaxE_{comp_i}} \tag{4.2}$$

$$W_{comp_i} = \frac{MaxE_{comp_i}}{\sum_{i=0}^{n} MaxE_{comp_i}} \tag{4.3}$$

As long as the smart device is powered on, the current application type and the current screen brightness are accessible from the operating system. The node cost is calculated according to this information and the record in the application profile table is updated as indicated above. If the running application has no record in the application profile table, AWERA monitors all the relevant readings as described in the previous section and creates a new application profile entry. The following description explains how the node cost is calculated based on the application profile.

In equation (4.1), G_{app} represents the utility function corresponding to the energy constraints imposed by the application on all the major device hardware components considered: CPU, screen, graphics, WLAN card and cellular module, respectively. G_{comp_i} represents the utility grade corresponding to the energy constraint imposed by the applications on the i-th device component. Normalized weights are used to balance the contribution of different hardware components on the overall utility function. For example, WLAN interface card shows significantly higher energy consumption of up to 7 times the sum of the other hardware components energy consumptions in highly network-intensive applications [9]. Weight values W_{comp_i} are obtained by dividing the maximum energy consumption of each of the components to the maximum system energy consumption as indicated by equation (4.3), where $MaxE_{comp_i}$ represents the maximum energy consumption of the hardware component i. For each application scenario, G_{comp_i} of each individual component is obtained according to the ratio of typical energy consumption (E_{comp_i}) over the maximum energy consumption of that component ($MaxE_{comp_i}$), as described by equation (4.2).

In Table 4.1, G_{CPU}, G_{GRA}, G_{CELL} and G_{WLAN} are the utility grades of actual components of G_{app} and the table demonstrates how equation (4.1) can be applied for different applications. For application j, $E_{CPU}(j)/MaxE_{CPU}$ gives G_{CPU} which is given by the value of the typical CPU energy consumption in normal situations when the current application is running over the maximum CPU energy consumption. $E_{GRA}(j)/MaxE_{GRA}$, $E_{CELL}(j)/MaxE_{CELL}$ and $E_{WLAN}(j)/MaxE_{WLAN}$ follow the same principle.

Table 4.1 illustrates the structure of the application profile. In the *Application Profile Table*, AWERA records the typical workload of each component. Table 4.2 shows one example of such a record [8].

$$E_{frac} = \frac{E_{cons}}{E_{total}} \tag{4.4}$$

$$G_{eLevel} = F_b \cdot \frac{E_{frac}}{e^{1-E_{frac}}} \tag{4.5}$$

As to the hardware specifications, AWERA deals with the variation proposed by all the major hardware components in the above. However, the battery characteristics and remaining energy level of the battery have to be considered. In equation (4.4), E_{cons} is the amount of consumed energy at the time of measurement and E_{total} is the total capacity of the battery. E_{frac} is used to denote the ratio of E_{cons} over E_{total}. Equation (4.5) presents the utility function associated with the energy level at the device. The equation uses an exponential formula to address the fact that the less energy amount is at the device, the more critical the situation is. The value of G_{eLevel} is normalized as a grade ranging from 0 to 1. In addition, the energy depletion is not the same for all battery systems. A factor reflecting individual battery features (F_b) is also introduced to compensate for this effect. This factor varies from one manufacturer to another, but remains constant for the same battery.

4.4.2 Routing Mechanism

$$E_{dev} = E_{local} + E_{forward} \tag{4.6}$$

Equation (4.6) shows the two components of the energy consumption of a device in the context of routing in the ad hoc wireless communication environment. E_{dev} represents the energy depletion of a device, which is comprised of the energy consumption of local process, for example, the data processing and data transmission for the current device: E_{local}, and the energy consumption of data forwarding to other devices in order to support ad-hoc wireless data transmissions: $E_{forward}$. The energy consumption for the local process is affected by the context of the device usage. It is advisable to encourage light loaded devices to act as intermediate forwarding nodes in order to conserve energy of the heavily loaded devices. This prolongs the lifespan of the energy critical devices.

In terms of the routing mechanism, AWERA enhances the AODV protocol [168] adding the application energy-awareness already described and making use of AODV's on-demand features and distance vector mechanism. AWERA enhancements are as follows.

- AWERA reacts to the change of the energy-related utility value gathered from the application layer and MAC layer instead of to topology changes only. Due to the limited processing capability and battery life, mobile users tend not to work on an application as long as they can do on a desktop PC. A heavily loaded device may become a light loaded device minutes later. Therefore, AWERA adopts an adaptive approach: it regularly initiates new route updates to react to such context changes.
- AWERA employs a cross-layer approach allowing the network layer to use application layer and MAC layer information for context information collection, as shown in Fig.4.2 and enabling routing strategy adjustments to be performed accordingly.

4.4.3 Routing Cost Computation

$$\begin{aligned} C_{link} &= \frac{1}{RSSI} \\ &= \frac{signal\_strength\_threshold}{received\_signal\_strength} \end{aligned} \tag{4.7}$$

The routing cost of one route is comprised of both node cost and link cost. The node cost reflects the context-based node characteristics as described in the previous sections. The link cost is represented by received signal strength indicator (RSSI) as indicated by equation (4.7). The device will transmit via the link with better reception if multiple links are available. Equation (4.7) shows one implementation of RSSI, which is the value of signal strength threshold of successful packet receiving over the value of the received signal strength. The reason of using RSSI is because: a) it provides straightforward representation of link quality; b) it can be measured and quantified from devices. Because AWERA is designed to balance energy savings and performance, the link quality is considered while performing routing. The wireless link with good quality may result in lower error rate and less often re-transmission, which are desirable for wireless data transmission.

$$C_{node} = W_{app} \cdot G_{app} + W_{eLevel} \cdot G_{eLevel}$$

$$W_{app} + W_{eLevel} = 1 \tag{4.8}$$

$$\begin{aligned} C_{route} &= \sum_{i=0}^{n} \Big(W_{node} \cdot C_{node}^{i} + W_{link} \cdot C_{link}^{i} \Big) \\ & W_{node} + W_{link} = 1 \end{aligned} \tag{4.9}$$

Equation (4.8) puts together the utility functions proposed for the application-related energy drain and battery energy level. The utility cost of any individual device C_{node} is a normalized value in the [0 , 1] range influenced by two factors: G_{app} - the utility grade of the device according to the energy-related application layer information, and G_{eLevel} - the utility grade dependent on the remaining energy level. Normalized weights W_{app} and W_{eLevel} tune the contribution of each factor in the overall utility cost.

AWERA assumes each node within the wireless network deploys the proposed context-aware self-learning process and can compute the utility function as indicated by equation (4.8). Equation (4.9) presents how the cost of a route accumulates the utility cost of each node and each link along the path to give a total cost of the path C_{route}. AWERA selects the path with the least cost and stores it in the *Routing Table*. Notably, the weight values in equation (4.8) and equation (4.9) are to be tuned in each specific networking environment for optimal results.

4.5 Simulation Testing and Result Analysis - AWERA

4.5.1 Simulation Testing Environment

AWERA is tested in Networks Simulator 2, version 2.34 [1]. Three sets of simulation scenarios were built to test the proposed solution and compare AWERA with other routing solutions.

4.5.1.1 Brief introduction to NS-2

Network simulator version 2 (NS-2) is a discrete event network simulators. The core of NS-2 is comprised of the C++ implementations of networking protocols across five layers, including application, transport, network, MAC and physical layers, and hardware networking interfaces used for networking communications. The testing scenarios are described using OTcl programming language. The discrete event scheduling environment supports the execution of the testing scenario and produces trace files which records the events happened during the whole period of simulation.

[1] Network Simulator 2, http://www.isi.edu/nsnam/ns/

4.5.1.2 AWERA NS-2 Model

The complete implementation of AWERA in NS2 is distributed into 3 major classes. Fig.4.4 presents the class diagram of AWERA implementation in NS-2. "awera_rtable" is the class that manages the routing table specifically designed for AWERA. This table accommodates the dedicated data entries that hold the routing cost for each route. The second class "awera_rqueue" manages the buffer queue sitting in the network layer. The buffer size and buffering strategy can be tuned for different simulation purposes. "awera" is the key class while others only assist it to perform its function. "awera" can invoke the embedded energy model "energy_model" of the current node "MobileNode".

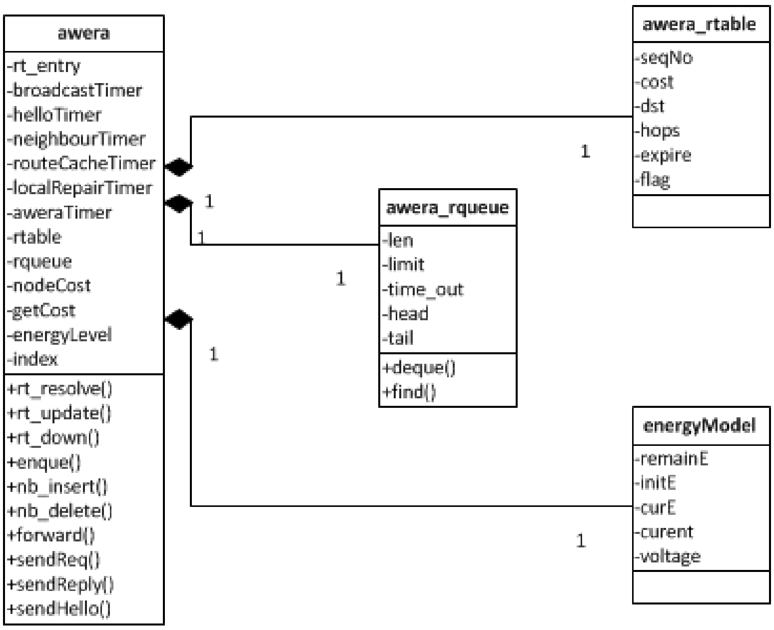

Figure 4.4: Class Diagram of AWERA Implementation in NS-2

The "energy_model" class maintains all the information about device energy source, including remaining energy, initial energy, current and voltage. According to the state of the wireless NIC, the wireless NIC module will decrease the corresponding amount of energy from the energy source. Consequently "awera" is able to obtain the current energy level. Regarding the energy level and energy depletion rate, the current energy constraint is calculated by dedicated method "getRoutingCost()" member of the "awera" class.

The route construction process takes an ad-hoc approach. When routing construction is needed, the "awera" class of the initiating node will call "sendRequest()" to broadcast a request for route. If a route exists, the receiving node will calculate the routing cost similarly to AODV except the energy-aware routing cost is considered for the existing route to the destination rather than the "number of hops". When receiving multiple replies, the one with the least energy-aware routing cost is selected and kept in the dedicated routing table "awera_rtable".

Importantly, every node has to broadcast the availability of itself and the change of energy level. Therefore "recvHello()" and "sendHello()" have been introduced. In this context, timers are configured to periodically invoke the corresponding functions for routing information update.

In conclusion, the "awera" class maintains energy-aware routing information, executes routines to initiate and perform route construction. Meanwhile it periodically sends message of "energy information" and "connectivity information" messages for routing table update.

4.5.2 Simulation Scenario 1 - Power Source and Application Type Differentiation

4.5.2.1 Test Scenario Goals

For the first scenario, a simplified version of AWERA is used to demonstrate its capability to differentiate nodes by their application type and power source, namely battery or mains. AWERA follows all the descriptions from chapter 3, except the metrics for device differentiation are power source and application type only. Two categories representing different remaining energy levels are main-powered and battery-powered for this simplified scenario. This is because main powered devices are considered to have very high energy levels.

AWERA is compared with AWERA_S, a simplified AWERA considering remaining energy level only and AODV [168] with regard to energy efficiency, delay and throughput in both stationary and random mobility scenarios. The simulation settings are described in details next.

4.5.2.2 Simulation Settings

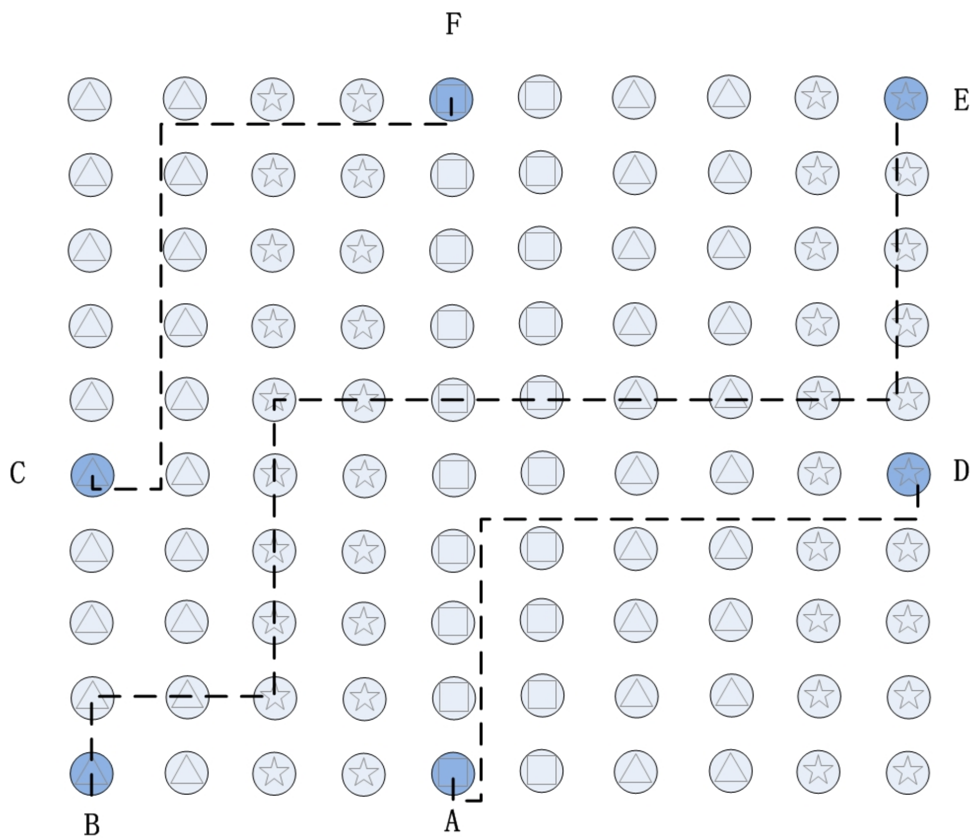

Figure 4.5: Simulation Topology - AWERA Scenario 1

A 115 m * 115 m grid topology of an ad-hoc WLAN comprised of 100 nodes located at 10 meter distance between each other is deployed as illustrated in Fig.4.5. 30 nodes represented with (triangles) are mains-connected and the others are battery charged. 40 battery-charged nodes (squares) are defined as running multimedia streaming applications and the other 30 battery charged nodes (stars) are defined as doing web browsing. The widely deployed IEEE 802.11g [85] is used as the MAC protocol. Tests are done in stationary and random mobile scenarios respectively. For random mobile scenarios, NS-2 random number generation program was used to generate random topology and random movement in terms of velocity and direction. 30 rounds of simulation were performed to obtain average simulation results.

With regard to energy constraint, transmitting power and receiving power are set as 0.035J and 0.031J respectively, the typical power values for Wifi network interfaces. The total energy capacity is set as 2J to reduce simulation duration on the server. And the energy capacity value will not affect the validity of simulation tests. The total time for each test is 400 seconds. The range of each node is defined as 30 meters.

Given the fact that networks are more and more involved in delivering rich media content, for example real time streaming of audio or video and VOIP, it is expected to perform well under such circumstances. Generated video-like constant bit rate data traffic of 1Mbps is used in all tests. Three flows of such traffic are considered (node A to node D, node B to node E and node C to node F). During the extensive tests, G_{app} is set to 0.1, 0.6 and 0.0 for web browsing, multimedia application, idle time respectively. G_{power} is set to 0.2 and 0.4 for mains connected and battery powered nodes, respectively, W_{app} and W_{power} are set to 0.5 to evenly distribute the influence of both factors. However, the weights and grades can be set differently. This is because the trends to be observed from the results are more important than the exact number of the results. Notably, this is a proof of concept based test. The above settings are summarised in Table4.3.

4.5.2.3 Simulation Results

In the first set of tests, stationary nodes are considered and the results are presented in Table 4.4. According to the results from Table 4.4, AWERA preserved 15.4% more

Table 4.3: Simulation Settings for AWERA Scenario 1

Parameter	Value
Number of Nodes	100
Node Distance	10m
MAC Protocol	IEEE 802.11g
Transmission Power	0.035J
Receiving Power	0.031J
Total Energy Capacity	2J
Simulation Duration	400 Sec
Video-like CBR	1 Mbps
G_{app} (web)	0.1
G_{app} (multimedia)	0.6
G_{app} (idle)	0.0
G_{power} (main)	0.2
G_{power} (battery)	0.4
W_{app}	0.5
W_{power}	0.5

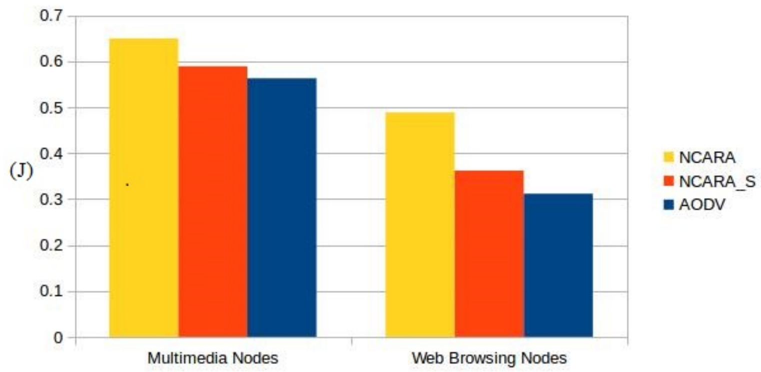

Figure 4.6: Average Remaining Energy for Nodes Performing Multimedia Delivery and Web Browsing, respectively - In Stationary Case (J)

energy than AODV for devices performing multimedia delivery, and 56.9% more energy for devices performing web browsing. On average the proposed solution achieved 91.7% of the throughput by AODV, 84% lower average delay than AODV. In Fig.4.6, the remaining energy values of multimedia delivery nodes are 0.63, 0.59, and 0.57 J for AWERA, AWERA_S and AODV. The difference is more pronounced for nodes performing web browsing. From Fig.4.7 and Fig.4.8, both AWERA and AWERA_S experienced a little degradation in terms of throughput an delay when compared with

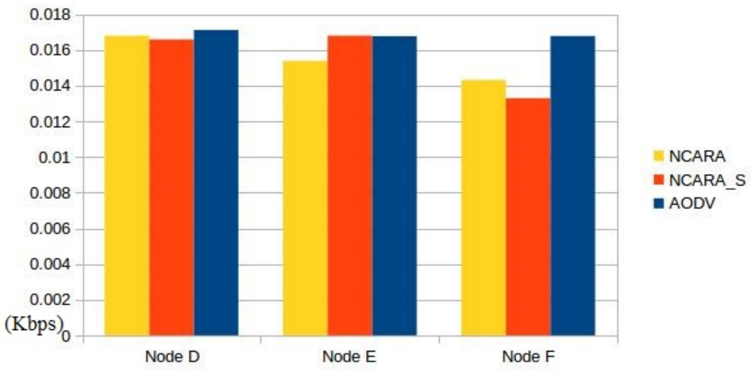

Figure 4.7: Average Throughput - In Stationary Case (Kbps)

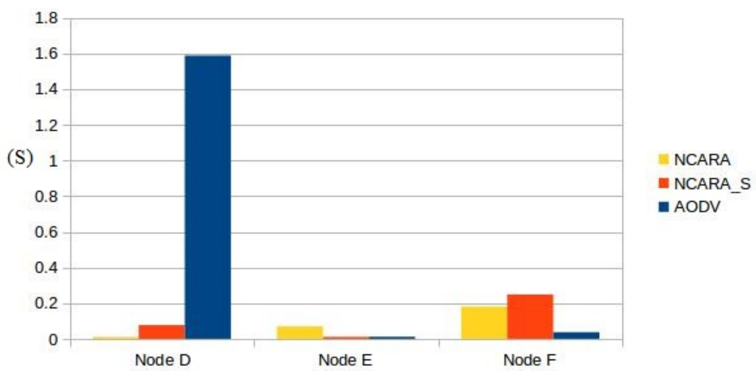

Figure 4.8: Average Delay - In Stationary Case (Sec)

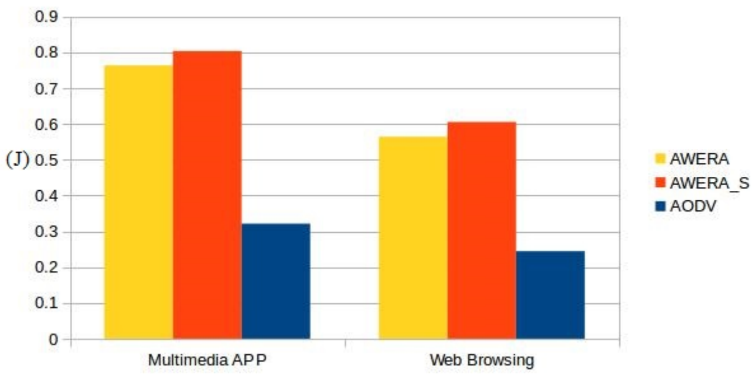

Figure 4.9: Average Remaining Energy for Nodes Performing Multimedia Delivery and Web Browsing, respectively - In Mobile Case (J)

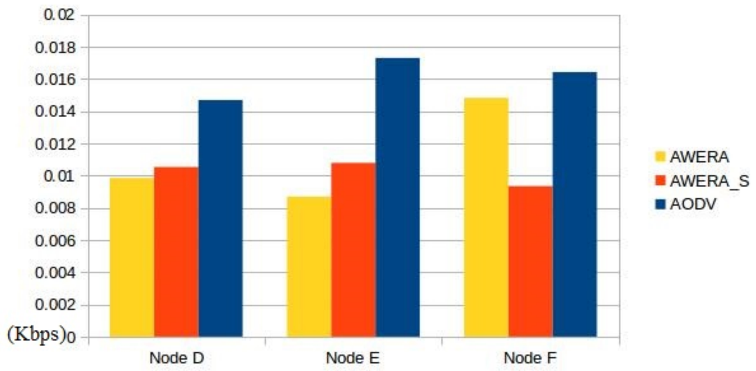

Figure 4.10: Average Throughput - In Mobile Case (Kbps)

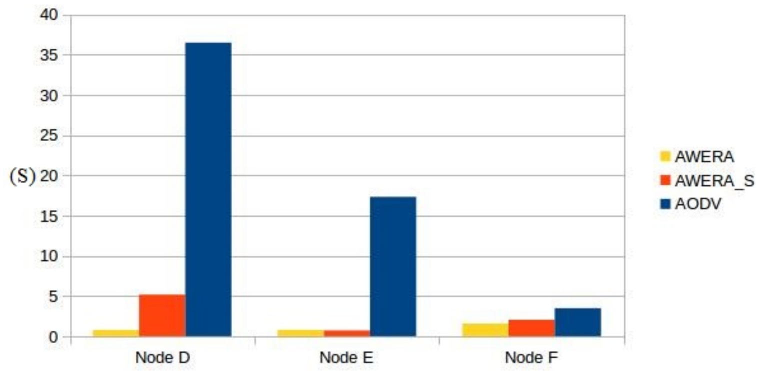

Figure 4.11: Average Delay - In Mobile Case (Sec)

Table 4.4: AWERA Scenario 1: Results from Stationary Scenario

	AWERA	AWERA_S	AODV	AWERA Benefit
remaining energy for streaming node (J)	0.650	0.589	0.563	15.4%
remaining energy for web browsing node (J)	0.489	0.362	0.311	56.9%
average throughput (Mbp)	0.016	0.016	0.017	-6%
Average delay (sec)	0.087	0.112	0.546	84%

AODV.

Table 4.5: AWERA Scenario 1: Results from Mobile Scenario

	AWERA	AWERA_S	AODV	AWERA Benefit
remaining energy for streaming node (J)	0.764	0.804	0.321	138%
remaining energy for web browsing node (J)	0.564	0.606	0.244	131.0%
average throughput (Mbp)	0.011	0.010	0.016	-31.1%
Average delay (sec)	1.042	2.645	19.110	94.6%

Mobility based testing results are presented in Table 4.5. From Table 4.5, it can be seen that AWERA outperformed AODV in terms of energy efficiency, conserving up to 134.7% more energy for multimedia delivery nodes and 131% more energy for web browsing nodes, in mobile scenario. This is as AWERA avoids suboptimal route and adopts periodically request mechanism, consequently superior results are obtained when dealing with mobility. This can be further demonstrated by the fact AWERA experienced around 5% of the average delay by AODV only. However, the AWERA throughput was reduced to 68.9% of that of AODV only. The advantages of AWERA is more pronounced for mobile cases. Fig.4.9 clearly shows that the remaining energy of multimedia nodes is nearly 0.8J and that of web browsing nodes is around 0.6J for AWERA, while the remaining energy for both kinds of nodes for AODV are below 0.35J. Fig.4.10 illustrates AODV still achieved highest throughput. From Fig.4.11, AWERA achieved shortest delay in mobile scenario while the delay of AODV is very high.

In conclusion, the proposed solution manages to differentiate nodes by their characteristics, including application type and power source, while making routing decisions, in order to conserve energy with little negative influence in terms of performance. Note that the nodes with hard energy constraints benefit more than other nodes when the proposed solution is used.

4.5.3 Simulation Scenario 2 - Light to Moderate Loaded Network

4.5.3.1 Test Scenario Goals

Full AWERA solution is being tested in light to moderate network conditions in this section. The routing cost of a route is comprised of both node cost and link cost. The node cost reflects the context-based node characteristics as described in the previous sections. The link cost is represented by RSSI (received signal strength indicator) as indicated by equation (4.7). The device will transmit via the link with better AWERA cost.

AWERA is deployed at each node within the wireless network therefore it can compute and accumulates the utility cost of each node and each link along a path to give a total cost of the path C_{route} as in equation (4.9). AWERA selects the path with the least cost and stores it in the *Routing Table* for use during the routing process. AWERA was compared with AODV and Minimum Battery Cost Routing (MBCR) [13]. MBCR chooses the total remaining battery capacity of every device along each path as the metric and gives the total remaining battery capacity of each route. It compares the battery capacity of all possible routes and uses the one with devices along it of most remaining energy.

4.5.3.2 Simulation Settings

A wireless network with an ad-hoc topology with 112 nodes was deployed at a 220m by 400m square grid. The distance between rows and between columns is 15m. For simplicity, all the nodes were classified into three classes based on the application type they are running: 38 idle nodes (class A), 37 gaming nodes (class B) and 37 video streaming nodes (class C). Since typically smart phone applications are designed to consume as little resources as possible, relatively low bit rate traffic was used in this simulations. Four pairs of randomly selected source-destination nodes transmitted

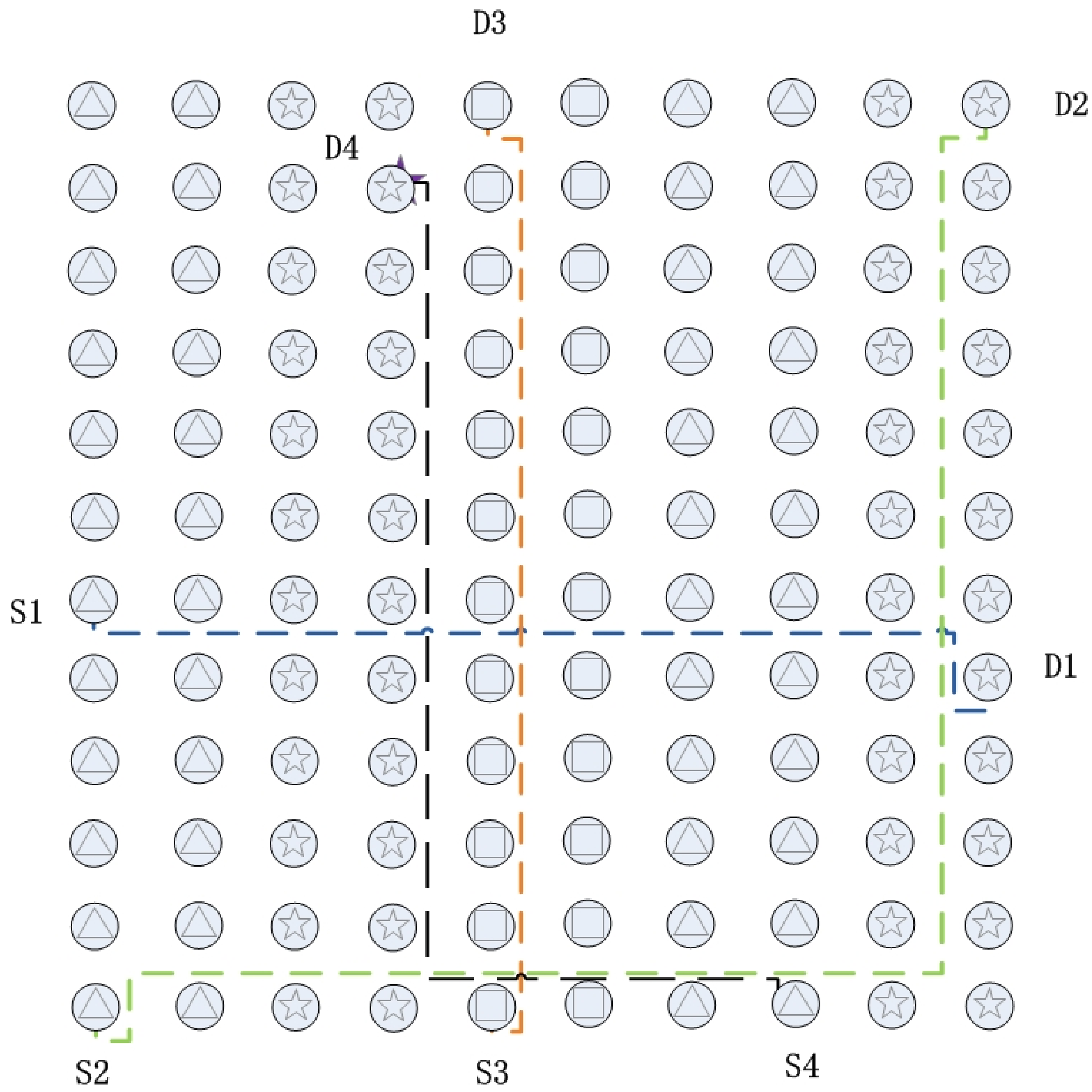

Figure 4.12: Simulation Topology - AWERA Scenario 2

video streaming-like traffic at three different qualities, at constant bit rates of 150 Kbps, 250 Kbps and 350 Kbps for three sets of tests. The simulation run for 30 rounds to obtain average results. Fig.4.12 illustrates the testing topology. However, the paths in the figure may not indicate the exact AWERA selected paths for routing in the actual simulations.

The duration of each of the simulations was 135 seconds. The average energy consumption per node for each group of nodes, average throughput in the four data streams, and average end-to-end delay were recorded. The simulations are conducted with AODV, MBCR [13] and AWERA, respectively and the results are compared.

The wireless network deployed the IEEE 802.11g standard and the transmission range for each node was set to 30 m. Idle nodes' power was set to 250 mW, gaming nodes' power was set to 450 mW and the power of the video streaming nodes was set to 650 mW. Additionally 400 mW was associated to the WiFi/WLAN transmissions. These settings are indicative values to reflect the difference of power demand for various kinds of applications. We defined G_{app} according to the typical power of each application scenario as indicated before. For the calculation of the energy application grade defined in equation (4.1), G_{app} was set to 0.25 for the class A nodes, 0.45 for the class B nodes, and 0.65 for the class C nodes. F_b in equation (4.5) was assigned a value of 1 in order to reflect the nature of the energy model in NS-2. When evaluating equation (4.8) W_{app} and W_{eLevel} are set to 0.5 to evenly distribute the influences. Since this is an evenly distributed topology, W_{link} and W_{node} in equation (4.9) are set to 0.1 and 0.9 in order to emphasise the importance of the energy consumption at the nodes. The initial energy for each node was 10 J, a reduced value in comparison with real life scenarios in order to reduce the simulation time.

The above settings are summarised in Table4.6.

4.5.3.3 Simulation Results

Fig.4.13, Fig.4.14 and Fig.4.15 illustrate how AWERA managed to conserve energy for each class of nodes when compared with both MCBR and AODV. As illustrated by Fig.4.13 and Fig.4.14, AWERA conserved 1.8J when compared with AODV. In Fig.4.15, AWERA achieved the energy saving of 0.4J when compared with AODV. In all the cases, MBCR performed in between AWERA and AODV.

Therefore regardless of the traffic rates, AWERA saved approximately 22 percent

Table 4.6: Simulation Settings for AWERA Scenario 2

Parameter	Value
Number of Nodes	112
Transmission Range	30m
MAC Protocol	IEEE 802.11g
Total Energy Capacity	10J
Simulation Duration	135 Sec
HQ Video CBR	350 Kbps
MQ Video CBR	250 Kbps
LQ Video CBR	150 Kbps
Idle nodes' Power	250 mW
Gaming nodes' Power	450 mW
Streaming nodes' Power	650 mW
WiFi Transmission Power	400 Mw
G_{app} (web)	0.45
G_{app} (multimedia)	0.65
G_{app} (idle)	0.25
G_{power} (main)	0.2
G_{power} (battery)	0.4
W_{app}	0.5
W_{eLevel}	0.5
W_{link}	0.1
W_{node}	0.9

energy for the class A nodes and between 11 and 15 percent energy for the class B nodes. The class C nodes spent much more energy in local applications than the other two type of nodes, and therefore not much energy savings can be made by employing a better routing strategy. Still AWERA conserved roughly 5 percent of the energy. These results show how AWERA considers the context of the applications and devices in the routing process and differentiates the energy savings based on it. In contrast, MBCR achieved only half the benefit offered by AWERA. Since the class C nodes consumed all their energy in t=135 seconds when using AODV, the simulations were stopped after 135 seconds for fair comparison.

AWERA also outperforms both AODV and MBCR in terms of performance, mostly due to the periodical updates and adaptive features of the energy aware routing. As illustrated in Fig. 4.16, AWERA experiences reduced end-to-end delay from 0.2 second to 0.5 second, depending on the case. MBCR achieved up to 33 percent lower delay

Table 4.7: AWERA Scenario 2: Results from Simulation (Data Rate: 150 kbps)

	AODV	MBCR	AWERA	Benefit over MBCR
Class A nodes: co-sumed energy (J)	6.813	5.86	5.075	13.4%
Class B nodes: co-sumed energy (J)	9.42	8.58	7.69	10.37%
Class C nodes: co-sumed energy (J)	10	9.84	9.595	2.6%
Average through-put (Mbp)	17.424	14.988	14.868	-0.8
Average delay (sec)	0.63	0.5	0.42	16%

Table 4.8: AWERA Scenario 2: Results from Simulation (Data Rate: 250 kbps)

	AODV	MBCR	AWERA	Benefit over MBCR
Class A nodes: co-sumed energy (J)	6.83	5.87	4.999	14.8%
Class B nodes: co-sumed energy (J)	9.40	8.59	7.602	11.5%
Class C nodes: co-sumed energy (J)	10	9.89	9.517	3.8%
average throughput (Mbp)	23.16	23.076	20.676	-10.4%
Average delay (sec)	0.67	0.62	0.24	61.3%

than AODV, but its performance was not stable when traffic rates increased. In return for the energy saving, the throughput of AWERA was up to 15 percent lower than that of AODV, while MBCR suffered less throughput degradation with the increase in the traffic rate, as it is shown in Fig.4.17.

In conclusion, AWERA enables energy saving with improved delay and little performance degradation in terms of throughput, outperforming the other two state-of-the-art routing solutions in terms of energy efficiency. More importantly, AWERA shows the ability to adapt to the context of wireless communications.

Table 4.9: AWERA Scenario 2: Results from Simulation (Data Rate: 350 kbps)

	AODV	MBCR	AWERA	Benefit over MBCR
Class A nodes: co-sumed energy (J)	6.971	5.9	5.314	9.93%
Class B nodes: co-sumed energy (J)	9.378	8.7	7.938	8.76%
Class C nodes: co-sumed energy (J)	10	9.875	9.699	1.78%
average throughput (Mbp)	27.9	27.132	24	-11.54%
Average delay (sec)	0.63	0.33	0.128	61.21%

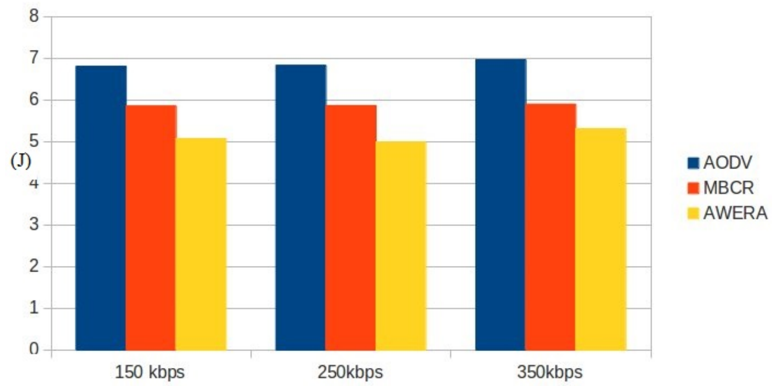

Figure 4.13: AWERA Scenario 2: Average Energy Consumption of Nodes from Class A (J)

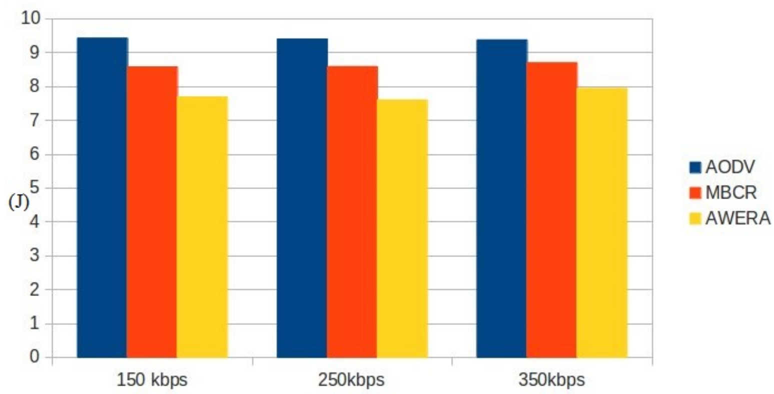

Figure 4.14: AWERA Scenario 2: Average Energy Consumption of Nodes from Class B (J)

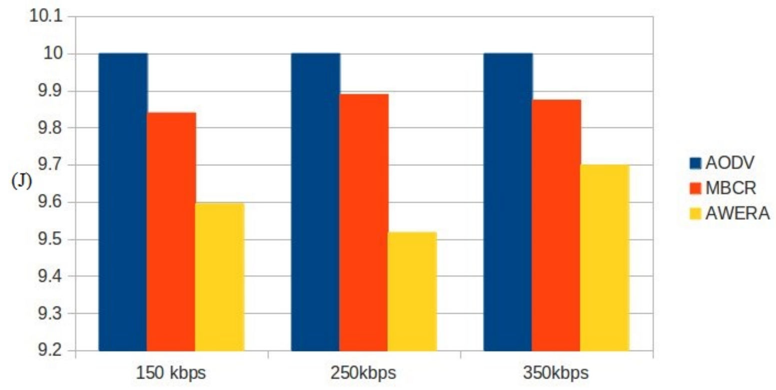

Figure 4.15: AWERA Scenario 2: Average Energy Consumption of Nodes from Class C (J)

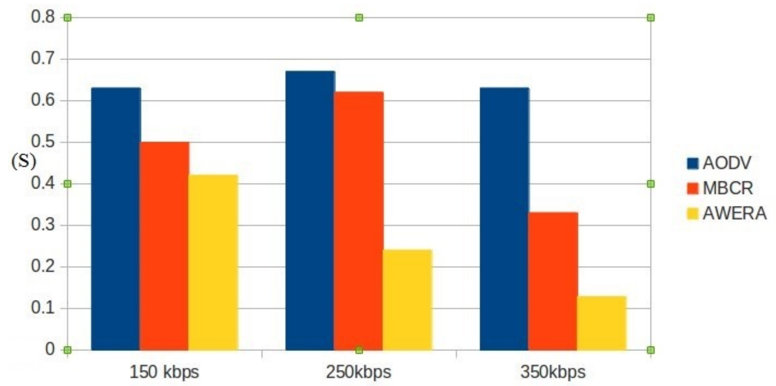

Figure 4.16: AWERA Scenario 2: Average End-to-end Delay (Sec)

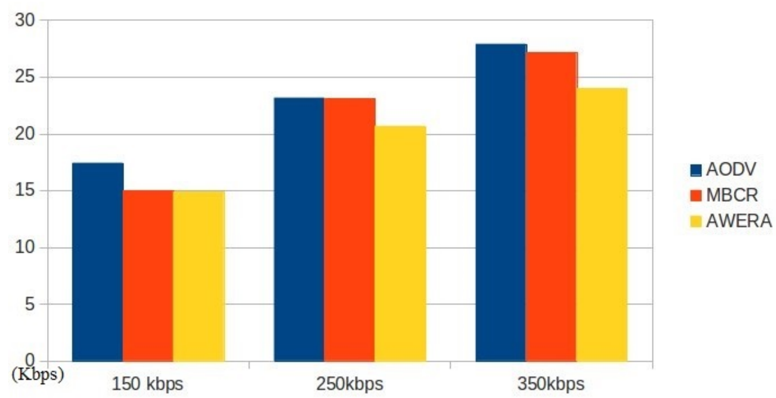

Figure 4.17: AWERA Scenario 2: Average Throughput of Receiver Nodes (Kbps)

4.5.4 Simulation Scenario 3 - Heavy Loaded Network

4.5.4.1 Test Scenario Goals

In this scenario, AWERA is being tested in heavy loaded networks, the routing cost of a route is comprised of both node cost and link cost. The node cost reflects the context-based node characteristics as described in the previous sections.

AWERA is deployed at each node within the wireless network therefore it can compute and accumulates the utility cost of each node and each link along a path to give a total cost of the path C_{route} as in equation (4.9). AWERA selects the path with the least cost and stores it in the *Routing Table* for use during the routing process. As in the previous sub section, AWERA was compared with AODV and Minimum Battery Cost Routing (MBCR) [13].

4.5.4.2 Simulation Settings

In this section, the performance of AWERA is evaluated by using NS-2[2] and a 200m by 200m square ad-hoc network topology with 150 randomly distributed nodes. For simplicity, three types of nodes were considered, which differ in the application type they are running: idle nodes (class A), video playout nodes (class B) and video streaming nodes (class C) as shown in Fig.4.18. There are 50 nodes of each kind. Three pairs of source and destination nodes were randomly selected. They transmitted video streaming-like traffic with the packets size of 210 bytes at constant bit rate (CBR) of 1 Mbps. Random movement was applied to all the nodes. Fig.4.18 illustrates the above description.

The duration of simulation was 420 s and the simulation was conducted 10 times with different seeds. The average energy consumption per node for each group of nodes, average end-to-end delay among the three traffic flows and average throughput

[2] Network Simulator 2, http://www.isi.edu/nsnam/ns/

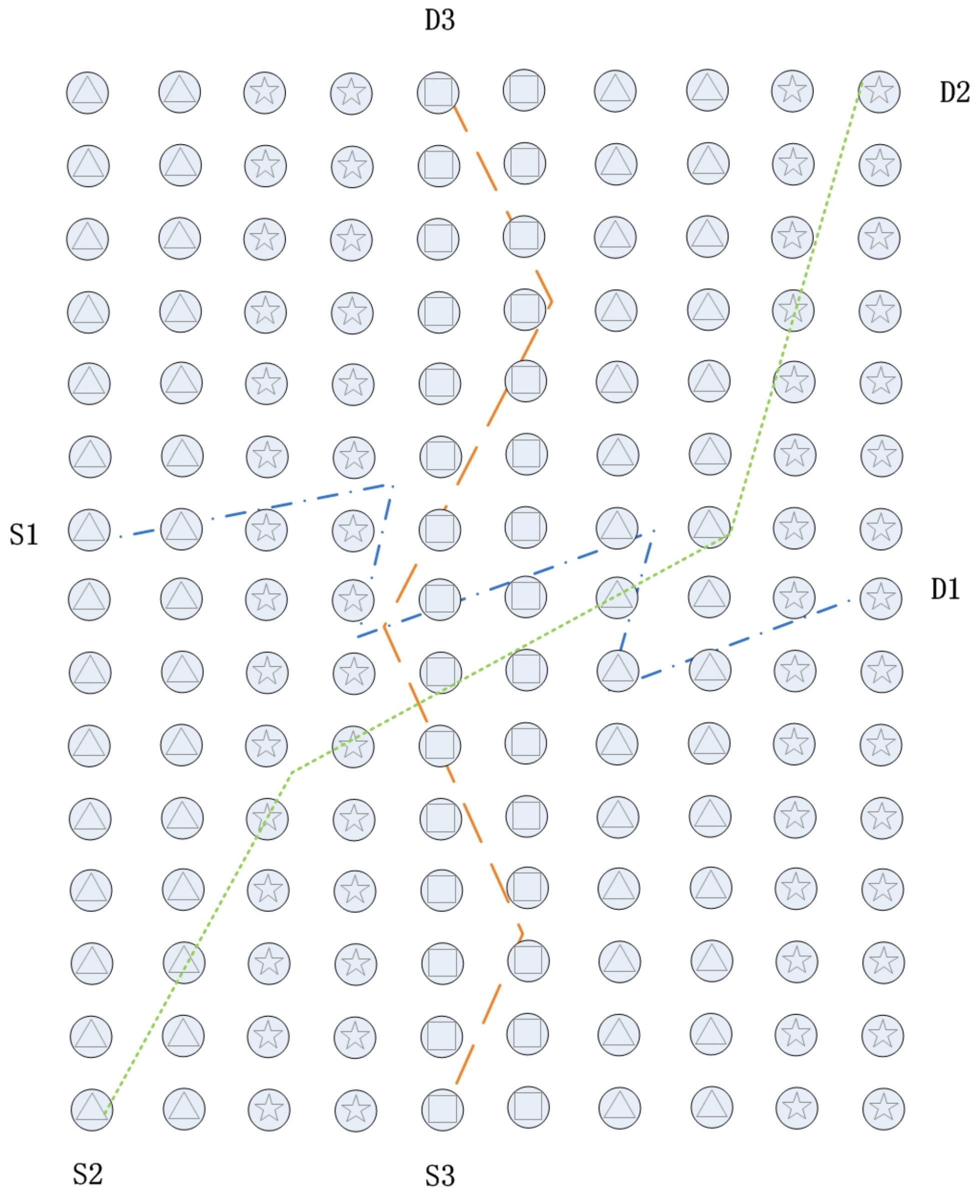

Figure 4.18: Simulation Topology - AWERA Scenario 3

in the three traffic flows were recorded. The simulation is conducted with AODV, MBCR and AWERA in turns and the results are compared.

IEEE 802.11b was deployed to the whole network. Transmission range for each node was set to 30 m. Idle nodes power was set to 250 mW, video play back nodes' power was set to 450 mW and the power of the 3G video streaming nodes was set to 650 mW. Additionally 400 mW was associated to the WiFi/WLAN transmission.

Table 4.10: Simplified model used for testing

Traffic Category	Idle	Video Playout	Video Streaming
G_{app}	0.25	0.4	0.95

These settings are indicative values to reflect the difference of power demand for various kinds of applications. The initial energy for each node was 30 J, a reduced value in comparison with real life scenarios in order to reduce the simulation duration.

In this scenario, we considered only two stages of energy consumption for each hardware components: full load (1) and minimum load (0). We defined G_{app} according to the typical power of each application scenario as above. For the calculation of the energy model as in equation (4.1), the grade of G_{app} was set to 0.25 for the class A nodes, 0.4 for the class B nodes, and 0.95 for the class C nodes. F_b in equation (4.5) was assigned a value of 1 in order to reflect the nature of the energy model in NS2 . From extensive tests, when evaluating equation (4.8), W_{app} and W_{eLevel} are set to 0.5 to equally distribute influences.

At simulation time t=420 seconds, the energy of the class C nodes was completely used when employing AODV. Consequently we recorded the average energy consumption per node every 100 seconds and calculated end-to-end delay and average throughput for 420 second simulation time, in order to be able to fairly compare the three solutions.

The above settings are summarised in Table4.11.

4.5.4.3 Simulation Results

The lifespan of all nodes were extended as illustrated in Fig.4.19, Fig.4.20 and Fig.4.21. Fig. 4.19 clearly shows how AWERA outperformed both AODV in conserving energy for each group of nodes with improvements of 10 percent to 20 percent. During the whole course of battery depletion, AWERA demonstrated increased benefit in terms of energy consumption in comparison with AODV and MBCR. As AWERA makes use

Table 4.11: Simulation Settings for AWERA Scenario 3

Parameter	Value
Number of Nodes	150
Transmission Range	30m
MAC Protocol	IEEE 802.11g
Total Energy Capacity	30J
Simulation Duration	420 Sec
Video-like CBR	1 Mbps
Idle nodes' Power	250 mW
Gaming nodes' Power	450 mW
Streaming nodes' Power	650 mW
WiFi Transmission Power	400 Mw
G_{app} (web)	0.45
G_{app} (multimedia)	0.95
G_{app} (idle)	0.25
G_{power} (main)	0.2
G_{power} (battery)	0.4
W_{app}	0.5
W_{eLevel}	0.5

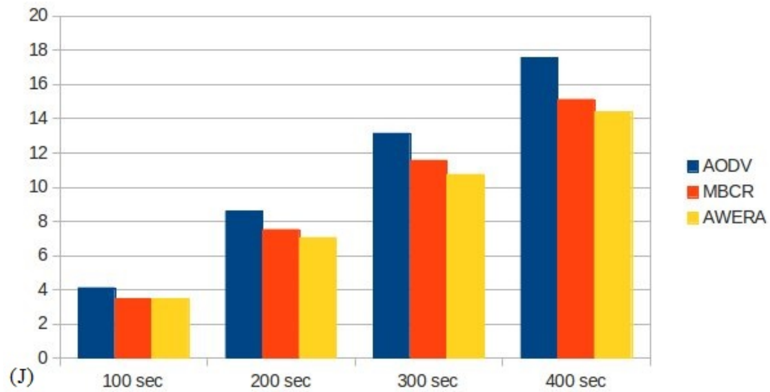

Figure 4.19: AWERA Scenario 3: Average Energy Consumption of Class A Nodes (J)

of adaptive application-aware information and differentiates its treatment of the nodes accordingly, it manages to conserve more energy when nodes suffer from more critical energy constraint. AWERA conserved 5 percent more energy than MBCR. Importantly AWERA managed to outperform MBCR considerably in terms of performance. Fig.4.20 illustrates similar results for class B nodes. Fig.4.21 demonstrates AWERA's benefit by up to 2J for class C nodes.

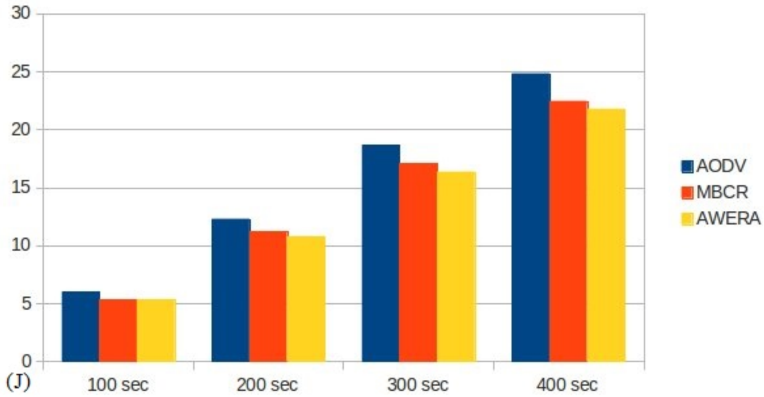

Figure 4.20: AWERA Scenario 3: Average Energy Consumption of Class B Nodes (J)

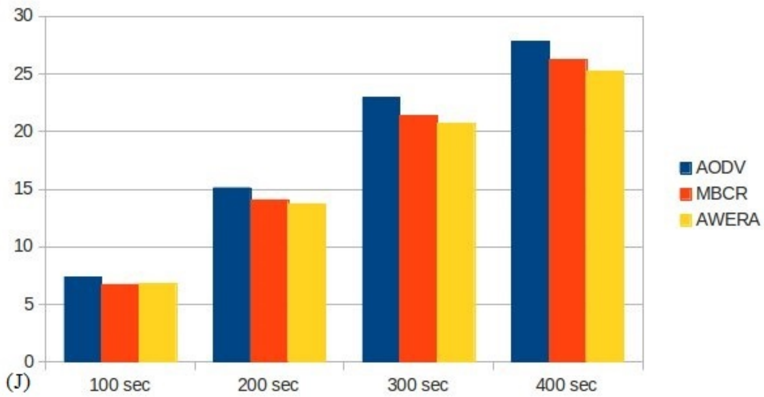

Figure 4.21: AWERA Scenario 3: Average Energy Consumption of Class C Nodes (J)

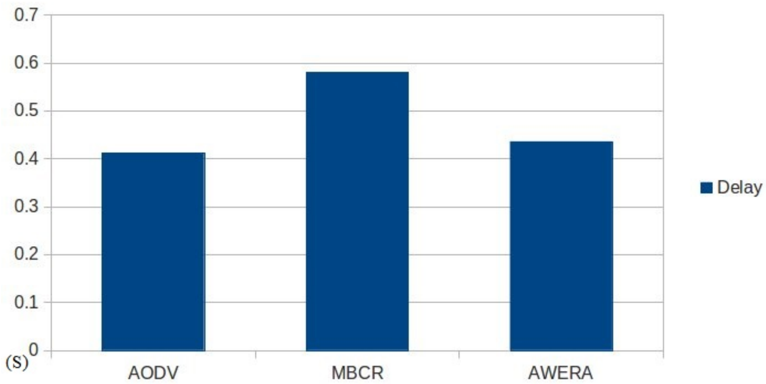

Figure 4.22: AWERA Scenario 3: Average End-to-end Delay (Sec)

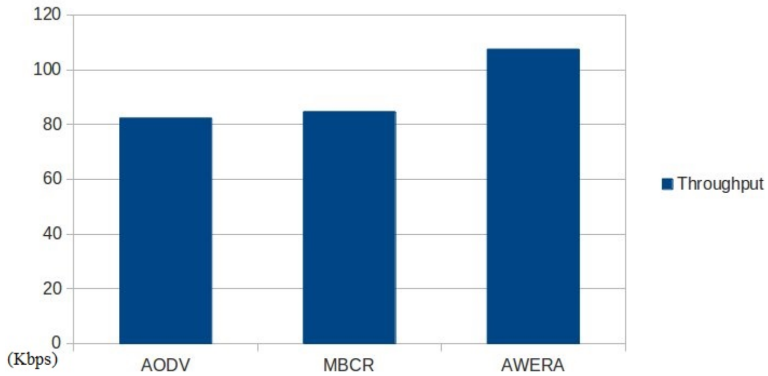

Figure 4.23: AWERA Scenario 3: Average Throughput of Receiver Nodes (Kbps)

Table 4.12: AWERA Scenario 3: Results from Simulation (1Mbps)

	AODV	MBCR	AWERA	AWERA Benefit over MBCR
Class A nodes: co-sumed energy at 400 sec (J)	17.562	15.116	14.395	5%
Class B nodes: co-sumed energy at 400 sec (J)	24.815	22.441	21.763	3.1%
Class C nodes: co-sumed energy at 400 sec (J)	27.810	26.232	25.236	3.94%
average throughput (Mbp)	82.3	84.6	107.4	26.95%
Average delay (sec)	0.412	0.58	0.435	33.33%

The end-to-end delay results are illustrated by Fig.4.22. With respect to the performance evaluation, MCBR experienced higher end-to-end delay than AODV by 0.17 s, in contrast AWERA increased the end-to-end by only 0.02 s. Fig.4.23 demonstrates that AWERA achieved higher throughput in comparison with AODV and MBCR. Compared with AODV, AWERA increased the throughput by 30.5 percent while MCBR has nearly the same throughput. This shows that AWERA is more competent in delivering multimedia content on time than MBCR, and AWERA offers improvement in throughput when compared with the other two solutions in heavy loaded networking environment.

In conclusion, AWERA achieved better energy efficiency than two other state-of-the-art schemes with much lower delay and very little degradation in throughput.

4.6 Chapter Summary

The energy constraint information produced by ASP process is utilised by AWERA. Utilising the above energy profiling process, the current device energy constraint is obtained in real time for path selection in a mobile device formed ad-hoc wireless networking environment. The routing cost computation, path selection algorithm, and the routing process are explained in details. First, this chapter presents the cross layer system architecture and describes AWERA. Then this chapter presents AWERA testing results obtained from three different sets of test. Three test beds were developed and different stage of the algorithm were tested.The models, settings of the simulated scenario, and results analysis are presented for each test scenario. These tests involve different traffic type, topology, mobility, and clearly show the benefits of adopting AWERA in comparison with the other routing solutions.

Chapter 5

Device-differentiated Adaptive Content Delivery Architecture, Algorithms and Testing

5.1 Overview

Data transmission and decoding are two most energy consuming tasks of video delivery. Video applications can easily reduce energy consumption by handling lower bit rate video sequences that require less data transmission and decoding effort. This is because the energy budget for video delivery at different quality levels vary significantly [169]. Therefore it is good practice to decrease the quality level when energy constraints become tight. Moreover, it is not even necessary for some small hand-held devices to receive too high quality video content, as the consequent improvement of user perceived video quality is very limited on those devices any way. Current mobile operating systems support multi-tasking, allowing multiple applications and background services to run on the machine simultaneously. Apart from the currently active video delivery application, the energy consumption of the other services on hold has to be considered. It is a very typical use case when the user opens a browser to send a post on Facebook, and then starts a Skype multimedia session, putting the browser on hold, without quitting. In energy constraint situations, content adaptation during the multimedia session can help reduce energy consumption.[151][144][143]

DEAS maintains an energy-oriented system profile in the device, and passively monitors all running applications. It also monitors the active video applications in terms of received data packets and calculates battery expected lifetime and perceived quality, as estimated using PSNR. The data collected in the system profile is used by an energy-efficient content adaptation algorithm to request the server to perform adjustments to the content delivery. This distributed approach reduces the computation load at the server, which is also a benefit of DEAS.

Then this chapter presents the simulation-based testing environment and simulation scenarios used to fully evaluate the performance of DEAS. For each scenario, the simulation settings and scenario description are described. The schemes used for performance comparison are introduced as well. The metrics used for assessment during the testing are presented along with the testing results.

The Client-Server video quality adaptation imposes overhead. The client needs to maintain perceived video quality and device energy constraint information. In addition, control packets need to be exchanged between the client and the server to enable quality adaptation. The typical RTCP packet size is 32bit. The total overhead is dependent on the frequency of quality adaptation. However, the quality adaptation result in substantial energy saving.

The overhead for a DEAS enabled device is computed as DEAS Control Packet Size * Update Rate. The Control Packet Size is 24 bytes in this case. The update rate is not fixed and dependent on the devices energy budget. DEAS enabled device only sends DEAS Control Packet when quality adaptation is needed. The overhead of ASP is also included in the overhead of AWERA. This is because AWERA uses ASP to obtain device energy constraint information.

5.2 DEAS Use Case Scenario

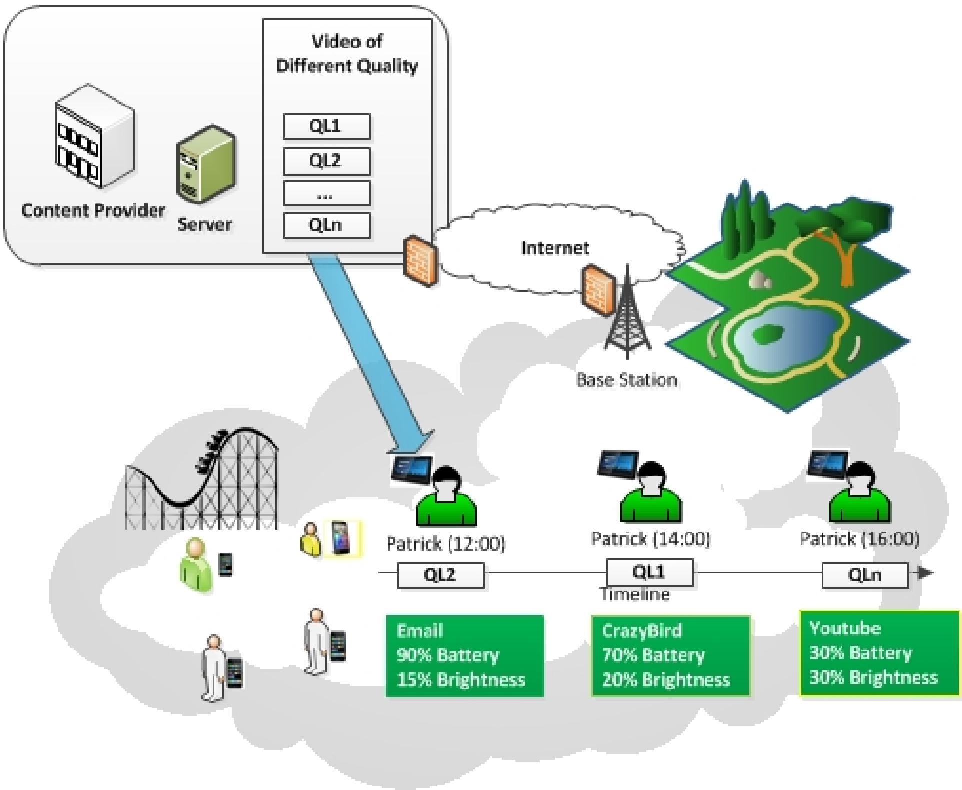

Figure 5.1: Use Case Scenario of DEAS

A typical deployment scenario of DEAS is depicted in Fig. 5.1. Patrick is watching a live wireless video streaming from his tablet in an amusement park. DEAS is in place for energy efficient adaptive video delivery from the content provider to Patrick. The energy constraint information retrieved from the energy profiling, video quality information, and PNSR scores are used to adapt the content quality to best balance the energy efficiency and user perceived quality. At 12:00 when Patrick has just arrived at the park, his tablet's battery is almost full and the screen brightness is low, which incurs less energy consumption. However, the network is slightly congested, which brings trouble for smooth delivery of high definition content. Consequently, the server from content provider side streams the video content at quality level 2 (QL2) quality to Patrick for smooth playback. At 14:00, only 20 percent of battery is drained, and the remaining energy is still plentiful. Besides, the network is not congested any more, and therefore the server decides to stream the content at QL1, the highest quality, to enable the best perceived quality possible. At 16:00, the remaining energy level is low and decoding and receiving high quality content is particularly energy consuming. At this stage, the content quality is decreased to prolong the device's battery life as much as possible. DEAS components are deployed both on the hand-held devices and the servers of content provider.

5.3 DEAS Architecture

The architecture of the proposed DEAS, the device-differentiated adaptive video delivery solution is presented in this section. Fig.5.2 illustrates the major components of DEAS and ASP, and the information sharing mechanism among them for energy efficiency in heterogeneous network environments.

In Fig. 5.2 *App Monitor* identifies the current running application each time a new application is launched. *Dev Info Collector* is responsible for fetching the status of the device from the *Mobile Operating System* by taking samples of multiple readings, in-

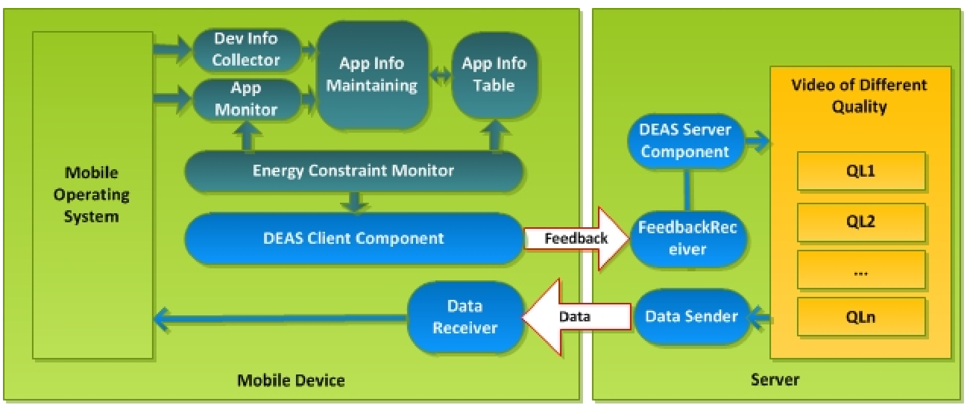

Figure 5.2: Block-level Architecture of DEAS

cluding work load on CPU, load of wireless network card, cellular interface, resolution of the display unit, battery characteristics, battery energy level. Based on the above information, *Energy Constraint Monitor* acts as the core of ASP to provide energy constraint information. Using the above information, *DEAS Client Component* evaluates the delivery quality of service, calculates expected battery life, and makes video quality level adjustment requests according to the DEAS's video quality adaptation algorithm. On the server side, *Feedback Receiver* listens for adaptation requests from the mobile device, and *DEAS Server Component* handles the requests by dispatching video at certain quality level (QL). Data packets are transmitted by the server's *Data Sender* to the *Data Receiver* at the mobile device.

5.4 DEAS Algorithm

DEAS involves the server to perform adjustments to the content delivery according to two metrics: battery expected lifetime, calculated based on the data collected from the energy-oriented system profile and perceived quality, measured using PSNR on the monitored active video application.

As revealed by extensive tests [169], the variation of traffic volume and decoding effort caused by video at different quality levels has a major impact on system energy efficiency when compared with that of link quality, network load and transport protocol. Consequently the proposed Energy-efficient Quality Adaptation Algorithm focuses on the above two metrics. The network load and link quality are also important factors, but this paper assumes homogeneity in terms of network aspects.

It is assumed that video content encoded at different rates and consequently at different quality levels (QL) is already stored at the server, or the server has means to adjust the video encoding rate on the fly at transcoding. The multimedia data is transmitted via RTP, and any DEAS enabled device transmits feedback via RTCP to the server. Based on the feedback, DEAS involves dynamic switching of the video content quality source and video delivery of an energy-aware adapted stream. This mechanism provides a user-centric energy-aware solution to address the challenges introduced by the heterogeneity of mobile devices in terms of software and hardware.

Expected battery lifetime is used to determine the maximum quality level upon which the adaptation is performed. Equation (5.1) presents the utility function to calculate the *expected_life* based on the currently running applications and individual device features. α_B is the compensation factor that reflects the depletion curve of the battery *B*. The $Residual_B$ is the residual energy of the battery recorded as current value, $Voltage_B$ is the voltage value of the battery used. They are both obtained from the mobile operating system. P_{sys} is the full load power of the system obtained in the Setup phase of ASP. G_{sys} is also obtained by ASP and it reflects the current energy constraints imposed by both the active application and applications on hold. The product of the formula is time, namely the expected battery life given the current energy constraint.

$$Exp\_Life = \frac{Residual_B * Voltage_B}{P_{sys} \cdot G_{sys} \cdot \alpha_B} \tag{5.1}$$

PSNR is used to estimate user perceived quality in DEAS. Based on PSNR, DEAS client makes adaptation requests to best balance quality of experience and energy saving. Equation (5.2) estimates PSNR [170], where *MAX_BitRate* is the maximum bit rate of the video, *Exp_Thru* is the expected throughput, and *Thru* is the actual throughput. When the PSNR value is below certain threshold, voluntary video quality level degradation is performed to save energy. This is because video content of lower quality requires less energy for decoding and transmission.

$$PSNR = 20 \cdot \log_{10}\left(\frac{MAX\_BitRate}{\sqrt{(Exp\_Thru - Thru)^2}}\right) [170] \tag{5.2}$$

DEAS performs the energy-efficient video quality adaptation as described in Algorithm 1. Since the mobile device receives the first data packet, it starts sending feedback regularly to the server. The proposed adaptation algorithm first calculates the expected battery life *Exp_Life*, RTT and PSNR for the current session. *Exp_Life* is compared with the threshold *Thre_Life*, indicating the desired session duration (e.g. video clip time length). If *Exp_Life* is shorter than the threshold *Thre_Min_Life*, the maximum quality level *Max_Level* is decreased to a lower level if any. If *Exp_Life* is longer than the threshold *Thre_Max_Life*, the maximum quality level is set to a higher level if any. Otherwise, it will keep the current quality level. Adaptation is caused by any energy-relevant event, for example, starting or ending an application, battery depletion in time, etc. If the computed PSNR value is less than the threshold *Thre_PSNR*, the device will also request a video quality level decrease via feedback. Otherwise, it will request a quality level increase from the server. The device waits for time-out before sending another feedback. The server performs quality adjustments following feedback requests. If no feedback is received by the server, it gradually reduces the video quality level to the minimum. This is because the network is assumed to be very congested.

However, the client transmitted request packets are considered overhead. Since the RTSP header length is 12 bytes and the time-out is configurable, the overhead is minimum in comparison with size of multimedia content, which can easily exceeds tens of hundreds of MB.

Algorithm 1 Energy-efficient Quality Adaptation Algorithm

```
Since the Smart Device Received the First Data Packet:
while True do
    Calculate(RTT, PSNR, Exp_Life);
    if (Exp_Life < Thre_Min_Life) then
        Request(Decrease, QL);
    else if (Exp_Life > Thre_Max_Life) then
        Request(Increase, QL);
    else
        Keep(QL);
    end if
    if (PSNR < Thre_PSNR) ∧ (QL > Lowest) then
        Request(Decrease, QL);
    else if (PSNR > Thre_PSNR) ∧ (QL < Max_Level) then
        Request(Increase, QL);
    end if
    Wait(Time_out);                ▷ Wait for Time_out before sending feedback
end while
```

5.5 Simulation Testing and Result Analysis - DEAS

5.5.1 Simulation Testing Environment

DEAS is simulated and tested in Networks Simulator 3, version 3.15 [1]. A heterogeneous networking simulation scenarios were built to test the proposed solution and compare DEAS with other solutions.

5.5.1.1 Brief introduction to NS-3

NS-3 is a replacement for NS-2. It has no backward-compatibility with NS-2. It is built using C++ and Python with scripting capability. Similarly to NS-2, the core of NS-3 is comprised of the C++ implementations of networking protocols across five layers, including application, transport, network, MAC and physical layers, and hardware networking interfaces used for networking communications. The testing scenarios are described using C++ or Python programming language. The discrete event scheduling

[1]Network Simulator 3, http://www.nsnam.org/

environment supports the execution of the testing scenario and produces trace files which records the events happened during the whole period of simulation.

In comparison with NS-2, NS-3's code tree is better organised with support for many new protocols contributed at every release. Moreover, NS-3 adopts modern design, pure contemporary C++ implementation and many handy features, for example, smart pointer and callback mechanism. Therefore, it is relatively easy to develop bug free implementation of networking protocols. The code is clearer to understand, which makes maintenance and further development much easier. Hence NS-3 is the choice for DEAS modelling and simulation.

5.5.1.2 DEAS NS-3 Model

The major components of DEAS are adaptive streaming client and adaptive streaming server. Fig.5.3 presents the class diagram of DEAS implementation in NS-3. To make an application-aware energy efficient adaptive delivery possible, an application-aware callback mechanism of WiFi energy model was implemented. Besides, NS3 provides a unique packet tagging mechanism, where add on information can be added to the packet head without inheriting and rewrite a new header. Therefore a dedicated tag class "seqPackt" is developed to accommodate relevant information: time stamp "tstamp", packet sequence number "seqNo", preferred quality level "ql". This information is used by both the client class "deas_client" and the server class "deas_server". The server listens for such information in order to delivery content of the preferred quality level. In function "getPSNRScore()", the client will calculate PSNR scores using this information. The client also uses the implemented application-aware energy modelling to obtain the current energy constraint score in function "getEnergyScore()". Based on the energy constraint score and PSNR score, the client calculates the demanded quality level in "calculateQL()" and sends the request packet containing such information to the server from within function "send()". When receiving video of different quality levels from the function "receive()", the energy model of the receiving device will set the battery discharge to the corresponding level for accurate simulation. This is made possible by the call back mechanism provided by NS3. The elegant design of NS3 requires helper classes, which are literally API interfaces to be used in the simulation script. Consequently, ”DEAS_client” and ”DEAS_server” classes require dedicated ”helper” classes to expose itself to the module users. Therefore ”DEAS_client_helper” and ”DEAS_server_helper” are implemented as API classes. The DEAS mechanism is implemented by 3000 lines of c++ code distributed in 12 C++ source code files.

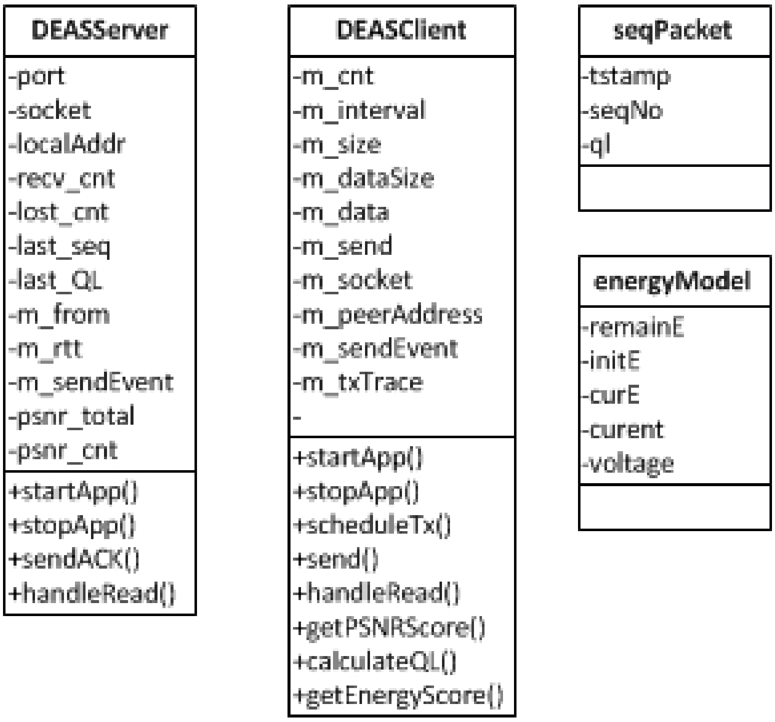

Figure 5.3: Class Diagram of DEAS Implementation in NS-3

5.5.2 Simulation Scenario 1 - Effect of Background Traffic Increase in A WiFi Network

5.5.2.1 Simulation Goals

e This test addresses DEAS' capability of saving energy while maintaining high QoS facing increasing background traffic of different type in heterogeneous wireless network environment.

5.5.2.2 Simulation Settings

The NS-3 simulation used the wired-cum-wireless topology illustrated in Fig.5.4 and the IEEE 802.11g wireless standard. The wired links had 100 Mbps bandwidth and 2 ms latency. Users A and B located initially near the west border are moving towards east with 0.5 m/s through the coverage area of the AP. The arrows in the figure illustrate the movement of the users. The distance between the users and the AP ranges from 50m to 70m. Server A stores video content at five quality levels and transmits video data packets to user A. Server B transmits UDP or TCP background traffic to user B. The background traffic differs in the six scenarios considered in the simulations as follows: A) 1 Mbps UDP traffic; B) 5 Mbps UDP traffic; C) 10 Mbps UDP traffic; D) 1 Mbps TCP traffic; E) 5 Mbps TCP traffic; F) 10 Mbps TCP traffic. This represents increasingly loaded networks and are meant to test the effect on video data delivery. The different background traffic classes are summarised in Table5.1.

Table 5.1: Background Traffic Classes for DEAS Scenario 1

Background Traffic Class	Bit Rate and Protocol
A	1 Mbps UDP
B	5 Mbps UDP
C	10 Mbps UDP
D	1 Mbps TCP
E	5 Mbps TCP
F	10 Mbps TCP

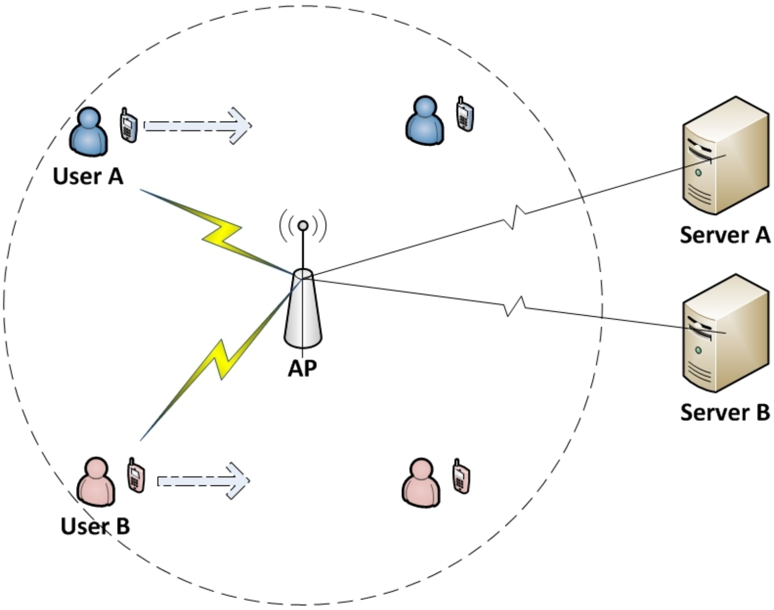

Figure 5.4: Simulation Topology - DEAS Scenario 1

DEAS is deployed for user A video delivery and follows the description from chapter 4. The compensation factor α_B in equation (5.1) is set to 1 to adhere to the linear depletion curve of the basic energy source model provided by NS-3. The battery voltage considered is 3.7 V, and the battery capacity is 186.48 J, 1% of the original value in order to reduce the simulation time until full depletion. *Thre_Life* and *Thre_PSNR* from Algorithm 1 are set to 100 s and 30 dB, respectively. The video content is of H.264/MPEG-4 format and stored in 5 quality levels with the average bit rates of 1920 kbps, 960 kbps, 480 kbps, 240 kbps and 120 kbps, listed from the highest to the lowest quality levels respectively.

In order to perform realistic simulations, the power settings are configured according to *Trestian et. al* [169], which performed comprehensive measurements with a Google Nexus One mobile device in a real test bed. They collected power readings when playing video content at different quality levels. Note that despite the tests being performed on the data gathered from a particular device, DEAS algorithm is still device independent. These power settings are 1445 mW (1920 kpbs), 1022 mW (960 kbps), 841 mW (480 kbps), 764 mW (240 kbps), 699 mW (120 kbps) for the five quality levels considered. The simulation duration is 100 s. After 80 s, a new application is launched which adds 100 mW power consumption to the system. This is to test system's reaction to both change of device features (battery residual level change) and applications.

DEAS is compared with a Non Adaptive video delivery solution (NonAd) and SAMMy [147]. NonAd does not perform any adaptation or quality adjustments. SAMMy adapts video quality based on link quality and energy level. SAMMy is a fair choice for comparison as it also performs video content quality adaptation, which is mainly how DEAS achieves its energy saving. The three delivery schemes are compared in terms of average values of packet loss rate and PSNR in order to illustrate their levels of quality of service and estimated user perceived quality, respectively. The performance of the solutions is assessed in terms of the device energy consumption.

The above settings are summarised in Table5.2.

Table 5.2: Simulation Settings for DEAS Scenario 1

Parameter	Value
Wired Link Bandwidth	100 Mbps
Wired Link Latency	2 ms
User Mobility Speed	0.5mps
Battery Voltage	3.7 V
Battery Capacity	186.48J
Thread of Battery Life	100 Sec
Thread of PSNR	30 dB
Bit Rate QL1	1920 kbps
Bit Rate QL2	960 kbps
Bit Rate QL3	480 kbps
Bit Rate QL4	240 kbps
Bit Rate QL5	120 kbps
Device Power (QL1)	1445 mW
Device Power (QL2)	1022 mW
Device Power (QL3)	841 mW
Device Power (QL4)	764 mW
Device Power (QL5)	699 mW
Simulation Duration	100 Sec

5.5.2.3 Simulation Results

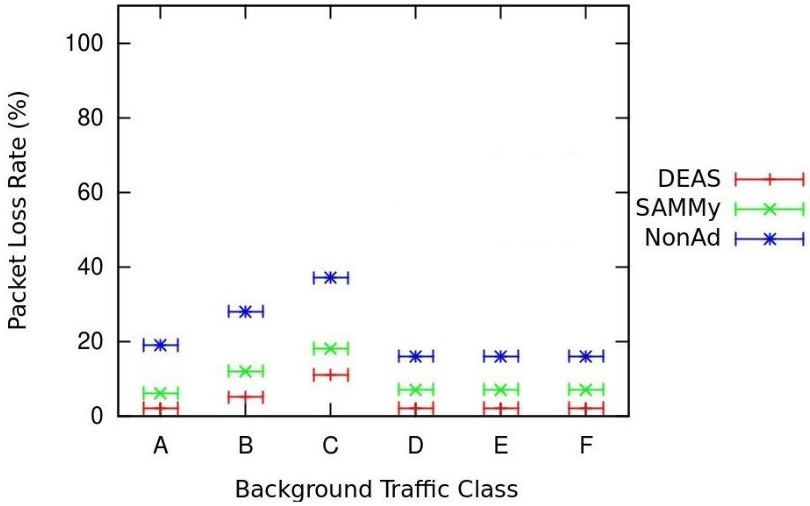

Figure 5.5: DEAS Scenario 1: Packet Loss Rate (%)

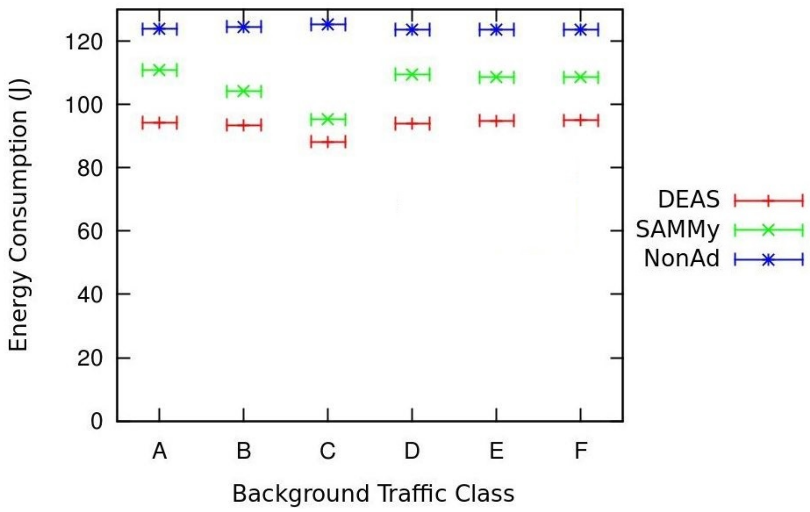

Figure 5.6: DEAS Scenario 1: Energy Consumption (J)

Fig.5.5 the packet loss ratios measured for the three schemes compared in the six different background traffic situations. It can be noted now the packet loss ratios of the adaptive schemes are much lower than those of the NonAd solution. DEAS's loss ratios are below 10% in all scenarios, roughly 5% less than those of SAMMy's. SAMMy suffers more from loss (∼20%) in heavy UDP background traffic conditions. NonAd's loss rate is nearly 20% worse than DEAS's values in all situations.

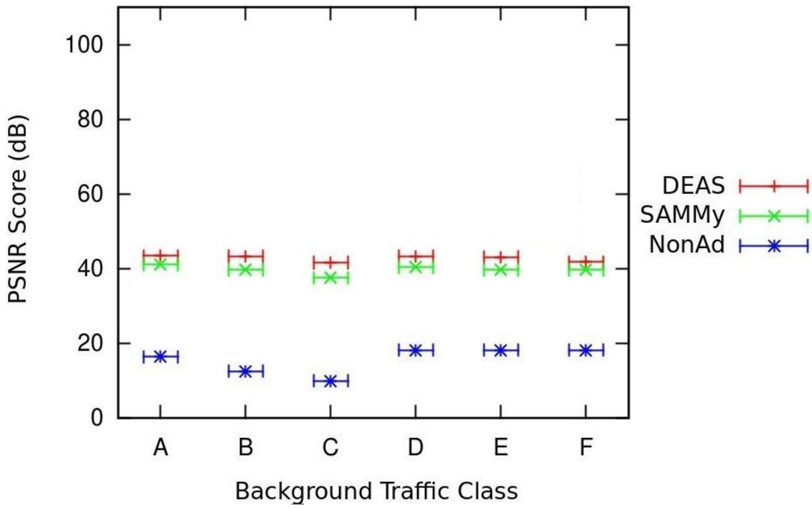

Figure 5.7: DEAS Scenario 1: Quality - PSNR (dB)

Fig.5.6 displays energy consumption for the three solutions with different background traffic. It can be noted how DEAS demonstrates significant benefits in terms of energy efficiency as illustrated in Fig.5.6. DEAS outperforms SAMMy by 10% and NonAd by nearly 30% in all TCP scenarios and lightly loaded UDP scenarios. DEAS's energy saving performance is even more evident in heavy UDP traffic conditions, when it records up to 30% benefit over SAMMy and 40% benefit over NonAd. For instance in case C, DEAS's energy consumption was 125J in comparison with 95J of SAMMy and 85J of NonAd.

In terms of estimated user perceived quality DEAS outperforms both SAMMy (by 5% on average) and NonAd (by more than 20 dB), as shown in Fig.5.7. The above results are presented in details in Table 5.3, Table 5.4 and Table 5.5.

Table 5.3: DEAS Scenario 1: Simulation Results for NonAd

NonAd	A	B	C	D	E	F
Loss Rate (%)	19	28	37	16	16	16
Energy Consumption (J)	123.77	124.47	125.21	123.54	123.54	123.54
PSNR (dB)	16.28	12.45	9.84	18.11	18.11	18.11

Table 5.4: DEAS Scenario 1: Simulation Results for SAMMy

SAMMy	A	B	C	D	E	F
Loss Rate (%)	6	12	18	7	7	7
Energy Consumption (J)	110.80	104.03	95.18	109.46	108.55	108.55
PSNR (dB)	41.04	39.64	37.47	40.39	39.72	39.72

Table 5.5: DEAS Scenario 1: Simulation Results for DEAS

DEAS	A	B	C	D	E	F
Loss Rate (%)	2	5	11	2	2	2
Energy Consumption (J)	94.155	93.433	88.201	93.866	94.821	94.913
PSNR (dB)	43.34	43.23	41.55	43.32	42.93	41.73

5.5.3 Simulation Scenario 2 - Effect of Background Traffic in a LTE Network

5.5.3.1 Simulation Goals

The goals of this test are to test the effect of network congestion of data transmission in LTE network to the performance of DEAS.

5.5.3.2 Simulation Settings

The topology used in this scenario is illustrated in Fig.5.8. Two servers streaming multimedia contents use LTE backbone to push data to the LTE gateway. The gateway relays the data further to the LTE base station, which provides coverage to all the ten content subscribers. The users are categorised into two classes: small icon (pink) users receive content from Server B; large icon (blue) subscribe for content from Server A. Server A stores video content at five quality levels and transmits data packets to users in pink. Server B transmits UDP traffic only.

Three devices subscribe for multimedia content from Server A. The number of subscribers of data from Server B differs in the six scenarios considered in the simulations as follows: A) 2 devices; B) 3 devices; C) 4 devices ; D) 5 devices; E) 6 devices; F) 7 devices. This is summarised in Table5.6. The growth of subscriber number result

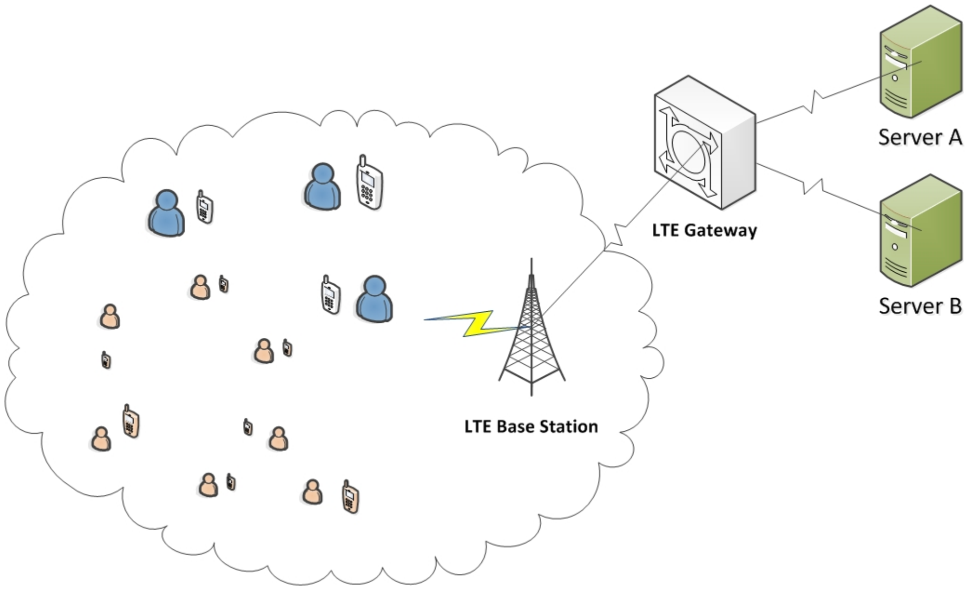

Figure 5.8: Simulation Topology - DEAS Scenario 2

in increasing the number of flows participating LTE transmissions in the cell. This test will find out how the compared solutions cope with the increasingly congested network environment. The data rate from video delivery from Server B is 2 Mbps. Random mobility at a walking pace is considered for all the mobile devices.

Table 5.6: Background Traffic Classes: Number of Background Traffic Subscribers for DEAS Scenario 2

Background Traffic Class	Number of Subscriber
A	2 Devices
B	3 Devices
C	4 Devices
D	5 Devices
E	6 Devices
F	7 Devices

The compensation factor α_B in equation (5.1) is set to 1 to adhere to the linear depletion curve of the basic energy source model provided by NS-3. The battery voltage is 3.7 V, and the battery capacity is 186.48 J, 1% of the original value in order to reduce the simulation time until full depletion. *Thre_Life* and *Thre_PSNR* from Algorithm 1 are set to 100 s and 30 dB, respectively. The video content is of H.264/MPEG-4 format and stored in 5 quality levels with the average bit rates of 1920 kbps, 960 kbps, 480 kbps, 240 kbps and 120 kbps, from the highest to the lowest quality levels.

In order to perform realistic modelling and simulations, the power settings are configured according to *Trestian et. al* [169], which performed comprehensive measurements with a Google Nexus One mobile device in a real test bed. They collected power readings when playing video content at different quality levels. These power settings are 1445 mW (1920 kpbs), 1022 mW (960 kbps), 841 mW (480 kbps), 764 mW (240 kbps), 699 mW (120 kbps) for the five quality levels considered. The simulation duration is 100 s. After 80 s, a new application is launched which adds 100 mW power consumption to the system. This is to test system's reaction to both change of device features (battery residual level change) and applications.

DEAS is compared with a Non Adaptive video delivery solution (NonAd) and an energy efficient adaptive delivery solution ESTREL [145]. NonAd performs no quality adaptation. ESTREL adapts video quality based on device's energy level and user perceived quality in order to save energy for devices. ESTREL is a reasonable choice for comparison as it also performs video content quality adaptation, which is mainly how DEAS achieves energy saving. The three delivery schemes are compared in terms of average values of delay, jitter and PSNR in order to illustrate their levels of quality of service and estimated user perceived quality, respectively. The performance of the solutions is assessed in terms of the device energy consumption.

The above settings are summarised in Table5.7.

5.5.3.3 Simulation Results

DEAS demonstrates significant benefits in terms of energy efficiency. Fig.5.9 presents DEAS outperforms ESTREL by 10% and NonAd by nearly 30% in all scenarios.

In terms of estimated user perceived quality DEAS outperforms both ESTREL and NonAd by up to 20 dB, as shown in Fig.5.10. In very congested network, DEAS

Table 5.7: Simulation Settings for DEAS Scenario 2

Parameter	Value
Wired Link Bandwidth	100 Mbps
Wired Link Latency	2 ms
User Mobility Speed	0.5mps
Battery Voltage	3.7 V
Battery Capacity	186.48J
Thread of Battery Life	100 Sec
Thread of PSNR	30 dB
Bit Rate QL1	1920 kbps
Bit Rate QL2	960 kbps
Bit Rate QL3	480 kbps
Bit Rate QL4	240 kbps
Bit Rate QL5	120 kbps
Background Traffic Bit Rate	2 Mbps
Device Power (QL1)	1445 mW
Device Power (QL2)	1022 mW
Device Power (QL3)	841 mW
Device Power (QL4)	764 mW
Device Power (QL5)	699 mW
Simulation Duration	100 Sec

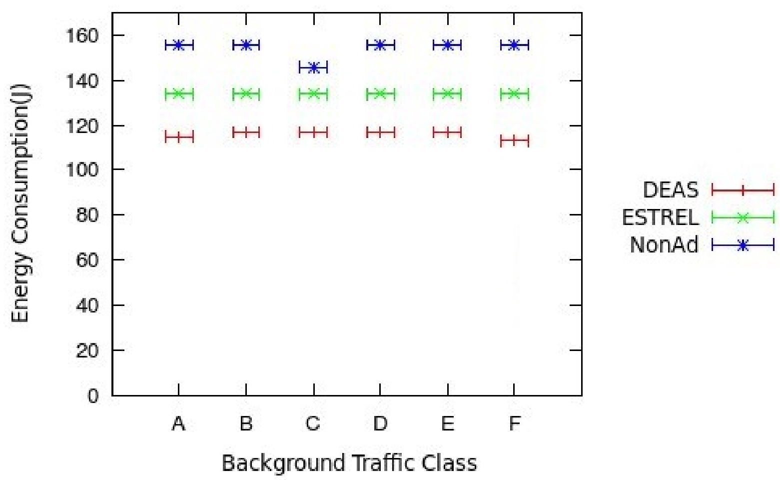

Figure 5.9: DEAS Scenario 2: Energy Consumption (J)

performed steadily whilst both NonAd and ESTREL experienced sever performance degradations. This is also clearly demonstrated in Fig.5.11 and Fig.5.12. The delay and jitter are significantly higher in Scenario D, E, F for both NonAd and ESTREL. This is because ESTREL was designed for energy saving and performance was not an

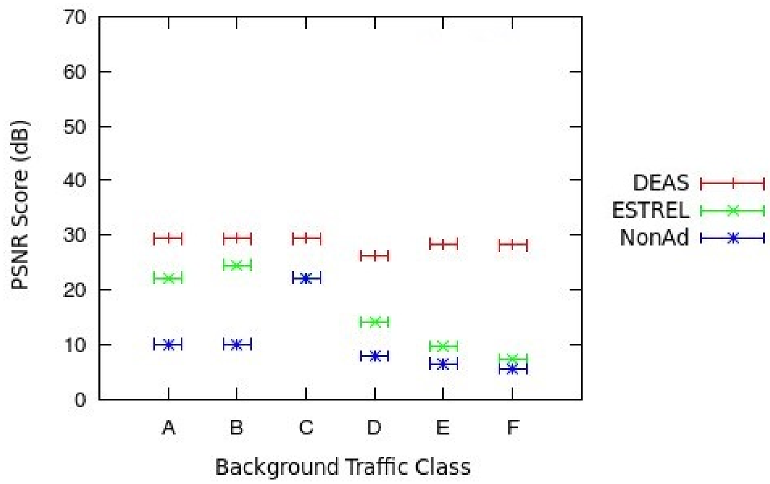

Figure 5.10: DEAS Scenario 2: Quality - PSNR (dB)

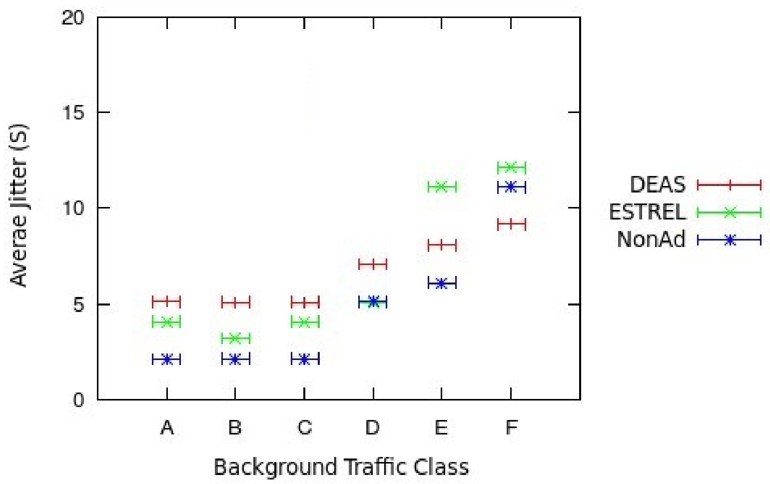

Figure 5.11: DEAS Scenario 2: Jitter (Sec)

issue and NonAd was not adapting anyway to increasing network load.

The above results are presented in Table 5.8, Table 5.9 and Table 5.10. These tests have shown how DEAS outperforms state of the art schemes in terms of performance and energy saving in both WiFi and LTE environments.

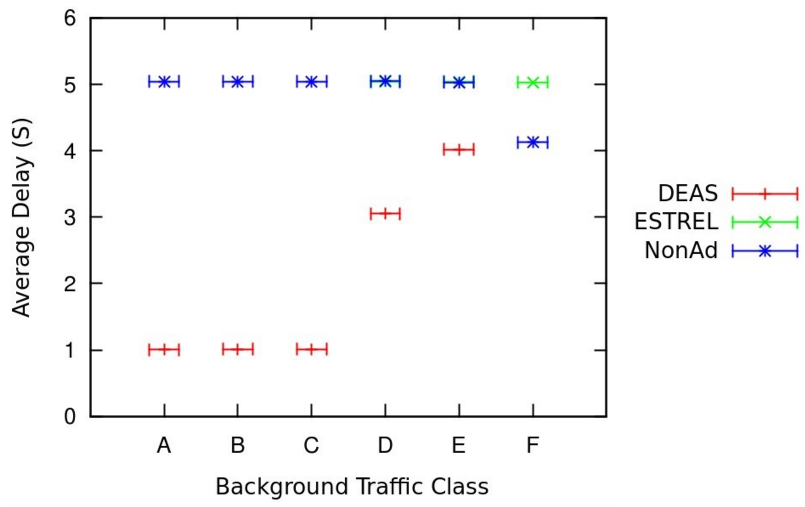

Figure 5.12: DEAS Scenario 2: Delay (Sec)

Table 5.8: DEAS Scenario 2: Simulation Results for DEAS

Background Traffic Class	A	B	C	D	E	F
Energy Consumption (J)	114.78	116.78	116.69	116.69	116.69	112.89
Delay (Sec)	1.0030	1.0100	1.0120	3.0500	4.0190	6.1410
Jitter (Sec)	5.095	5.070	5.079	7.080	8.060	9.168
PSNR (dB)	29.43	29.40	29.41	26.19	28.44	28.18

Table 5.9: DEAS Scenario 2: Simulation Results for ESTREL

Background Traffic Class	A	B	C	D	E	F
Energy Consumption (J)	133.73	133.73	133.73	133.73	133.81	133.73
Delay (Sec)	6.0792	6.0600	6.0792	5.0373	5.0420	5.0304
Jitter (Sec)	4.058	3.180	4.058	5.037	11.120	12.115
PSNR (dB)	22.19	24.57	22.19	14.14	9.67	7.22

Table 5.10: DEAS Scenario 2: Simulation Results for NonAd

Background Traffic Class	A	B	C	D	E	F
Energy Consumption (J)	155.61	155.62	145.75	155.61	155.52	155.61
Delay (Sec)	5.0408	5.0400	5.0400	5.0482	5.0300	4.1238
Jitter (Sec)	2.106	2.130	2.130	5.100	6.080	11.091
PSNR (dB)	10.08	10.09	22.17	7.89	6.50	5.59

5.6 Chapter Summary

This chapter presents the *Device characteristics-based differentiated Energy-efficient Adaptive Solution* (DEAS) for video delivery over heterogeneous wireless networks that performs content delivery adjustments with respect to the applications and device features in order to save device battery energy. DEAS makes use of the novel energy-oriented application-base device system profiling (ASP) which addresses challenges introduced by the heterogeneity of mobile devices. The benefits of DEAS are as follows. DEAS, as a new energy-efficient quality adaptation algorithm, targets not only the active video application, but also other applications and background services running on the same mobile device. In addition, ASP enables DEAS to perform low overhead accurate device differentiated quality adaptation on the fly based on the energy constraint. This energy modelling based quality adaptation has been little studied. Last, this chapter presents DEAS testing in different situations in comparison with other solutions. The scenarios involved including WiFi and LTE based heterogeneous wireless network environments, respectively. For each test, the goal and settings of the simulated scenario, and the results analysis are presented. Tests involving different traffic types and size clearly show the benefits of adopting DEAS over the other solutions.

Chapter 6

Energy-efficient Device-differentiated Cooperative Adaptive Multimedia Delivery Architecture, Algorithms and Testing

6.1 Overview

The increasingly powerful and affordable smart phones and tablet PCs play important roles in people's daily life all around the world. In airports, coffee shops and conference centres, many people incline to use such easy-to-carry devices for work and entertainment. In recent years, more than half of the Internet searches were performed on mobile devices[1].

Current smart devices are designed to be complex and compact, which requires them to be powered by batteries with limited lifetime. Consequently, the energy efficiency is a key issue in such scenarios. For multimedia content delivery, adaptive solutions that adjust content quality to network conditions have been proposed to improve energy efficiency while maintaining high QoS levels.

While watching video on mobile devices in places where population density is high, and user profiles might be similar, it is highly likely people nearby are requesting similar multimedia content. The existing various technology-based wireless networks and hot spots provided by businesses, public institutions, etc. establish a heterogeneous wireless network environment which can provide ubiquitous connectivity to smart devices. The different network interfaces (e.g. cellular, WiFi) of smart devices have different features in terms of energy efficiency and bandwidth support. Therefore sharing content between devices by utilising multiple network interfaces provides huge opportunity to improve the associated energy efficiency. In contrast, the existing content sharing schemes mainly focus on saving energy from the server side because the network traffic volume is reduced.

For content sharing in the above mentioned scenarios, there are several major challenges. The WiFi networks are often congested when the number of users is large; WiFi WLAN is not always accessible due to charging policy or limited availability of the hot spot. Secondly, devices with different energy constraints require content of different quality to prolong battery life. Most energy-oriented content sharing schemes [56][60][57] fail to address such issues.

6.1.1 Hybrid Content Sharing and Quality Adaptation for Energy Saving at Client Device

Most existing content sharing solutions focus on saving network bandwidth and easing the network resource on the content provider. Little effort has been put to improve the client device energy efficiency. Facing the deployment of heterogeneous wireless

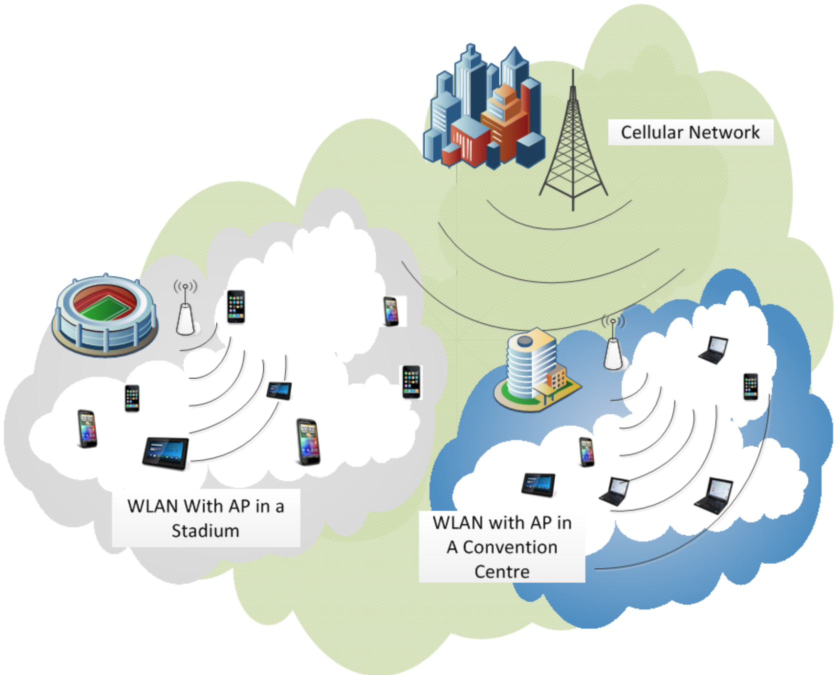

Figure 6.1: The Latest Heterogeneous Wireless Networks

networks that involve latest technologies, network (e.g, LTE, LTE-A, WiFi), the importance and potential of multiple network interfaces of smart devices is fully used. This is shown in Fig.6.1. The multiple network interfaces of current smart devices are not fully utilised for improving performance and energy efficiency. Each interface has its unique characteristics: for instance, WiFi interface has low energy-per-bit consumption with support of high bandwidth, the cellular interface has large coverage and good mobility support, and the blue tooth interface has low power demands for data transmission. If properly managed, content sharing using multiple interfaces can achieve good energy savings for smart devices.

For example, solutions like [56] organise neighbouring devices to download content for the device with less energy budget. However they fail to address the fairness of energy distribution among mobile devices. Moreover, the WiFi network may suffer from poor performance as a result of overcrowded users in a relatively small space. In fact, this is very common in public areas, where content sharing techniques are most likely to be used. Quality adaptation improves QoS by lowering the bit rate of the content when the network is congested; lower bit rate content requires less energy to transmit and decode.

In conclusion, the proposed hybrid solution, where quality adaptation is used in conjunction with content sharing, aims at both saving energy for the client device and improving QoS levels for multimedia delivery.

6.1.2 Device Profiling for Quality Adaptation and Partner Selection of Content Sharing

The existing energy saving solutions consider only the case when devices are working in the same context, and use metrics related to content delivery QoS, network conditions and energy level to adjust the delivery strategy. This assumption obviously does not perfectly suit current wireless communications scenarios, especially for the smart devices dominated heterogeneous wireless networks.

In such a scenario, all the applications running on the smart devices should be taken into consideration. This is because all the applications put various levels of energy pressure on the device. Along with the multimedia delivery application, multiple other applications could be running on the mobile device at the same time. A device performing multimedia streaming only is certainly less energy constraint than when it is running multiple applications. At the same time, two devices running the same application with different energy levels have different energy constraints. Therefore an intelligent content delivery strategy can improve energy efficiency by differentiating individual devices in different application load scenarios, and by adjusting the content quality accordingly.

This chapter presents a **Device-differentiated hybrid content sharing and quality adaptation solution** (EDCAM) for energy efficient multimedia content delivery over heterogeneous wireless networks to smart devices. Based on the current device energy constraint provided by ASP, EDCAM performs energy-aware quality adaptation using DEAS. In addtion, EDCAM allows two devices cooperatively download the same multimedia content for further energy saving. The content sharing and quality adaptation are performed based on the device characteristics.

Most existing content sharing schemes use ad-hoc networks to form sharing groups, where much effort is put into group management. This offsets the energy saving benefit at the client side. Additionally, the throughput is degraded as well due to the use of multi-hop transmissions[58].

EDCAM uses only one neighbouring device for content sharing with which there is common interest on the content for content sharing. The partner selection is based on the energy-oriented device characteristics. A quality adaptation algorithm is adopted for further energy saving. While the existing quality adaptation algorithms are based on energy level and network conditions only, the proposed EDCAM adapts video quality based also on the device characteristics.

EDCAM uses ASP to record power signatures of various device components for each running application in order to construct the device energy profile. Based on this profile: a) if neighbouring devices sharing the same interest on the content are available, the devices with similar energy constraint are selected as partners for content sharing; b) an energy efficient content adaptation is performed for the content delivery.

This chapter presents EDCAM in detail, including its architecture, energy-oriented system profiling, content sharing with the partner device, and energy efficient content adaptive delivery mechanism.

Then this chapter presents the simulation-based testing environment and simulation scenarios used to fully evaluate the performance of EDCAM. The simulation settings and scenario description are described. The schemes used for performance comparison are introduced as well. The metrics used for assessment during the testing are presented along with the testing results.

Both the content sharing and the quality adaptation introduce overhead. The overhead of quality adaptation has been discussed in the previous chapter of DEAS. The overhead of content sharing is mainly introduced in the connection establishment stage. First, the initiator needs to broadcast the content interest. Facing multiple replies, the initiator needs to calculate and select the optimal sharing partner. After which, the three way handshake took place. The control packets need to be exchanged to negotiate the order and size of segments to be shared. Once the sharing happens, the energy saving is substantial. However, it is also possible that no reply is received after the broadcasting stage. In that case, the three way handshaking will not happen though.

The overhead of EDCAM include the overhead of ASP, DEAS and the overhead of establishing sharing partnership. The overhead of establishing sharing partnership is computed as broadcasting packet size + response packet size * number of response packets + INIT packet size + SYN packet size + ACK packet size. The broadcasting packet has the content interest information. The node may receive multiple response packets. Once a partner has been selected, it sends INIT packet, receives SYN packet and sends ACK packets sequentially to perform three way hand shake. The above packets are configurable, and the typical packet size is configured as 52 bytes in this case. Hence the total overhead is dependent on the number of responses.

6.2 EDCAM Use Case Scenario

A typical scenario of EDCAM usage is illustrated in Fig.6.2. Imagine Patrick watching a live video streamed from a network on his tablet in a cellular network in an amusement park. As EDCAM is deployed, EDCAM enables devices to pair up and cooperatively download the content. In this manner, each device will download half of the content over the expensive and energy consuming cellular network, and retrieve the other half from its sharing partner device via the free and energy efficient WiFi network. As usual, the partner selection and content quality adaptation are based on

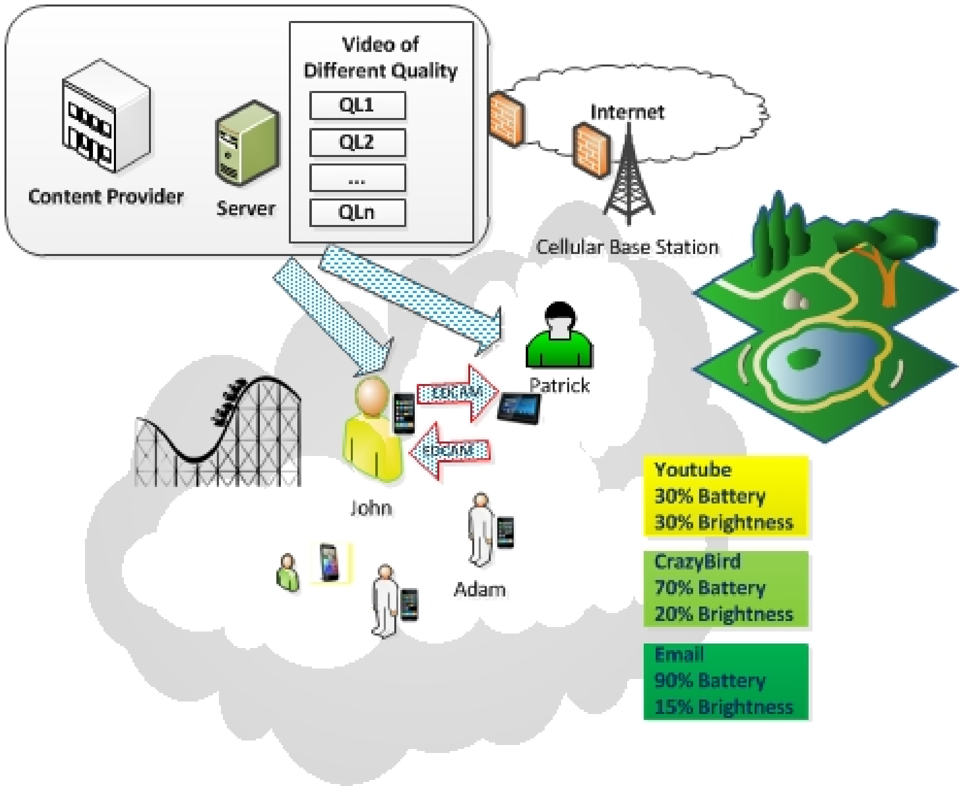

Figure 6.2: Use Case Scenario of EDCAM

device energy constraint. In Fig.6.2, Patrick teams up with John to cooperatively download the video and share the content with each other. John is chosen as he suffers from even tighter energy constraint compared with Adam. Additionally, quality adaptation is performed during content delivery, similar to DEAS.

6.3 EDCAM Architecture

The following subsections introduce the architecture of the three major elements of EDCAM: energy-oriented system profiling; energy-efficient content adaptive delivery mechanism and cooperative content delivery with a partner device.

Fig.6.3 illustrates the EDCAM-based multimedia content delivery. In the traditional approach, each device gets the whole content via the same network interface (e.g. cellular), independent from other devices. The new approach of EDCAM en-

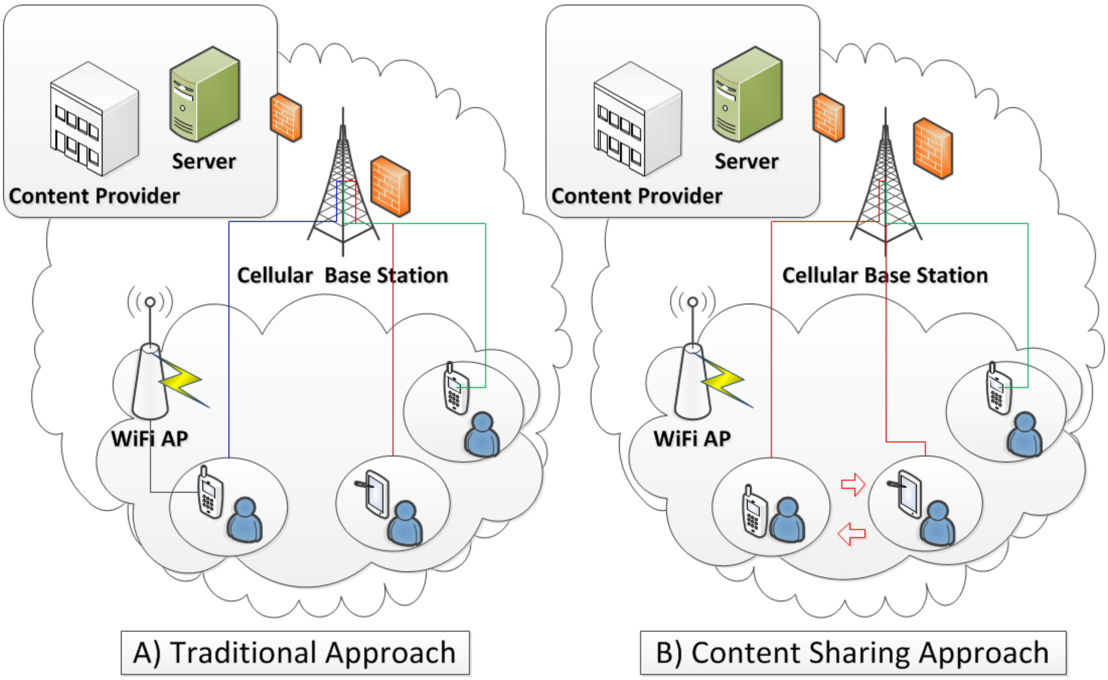

Figure 6.3: Different Multimedia Content Delivery Strategy

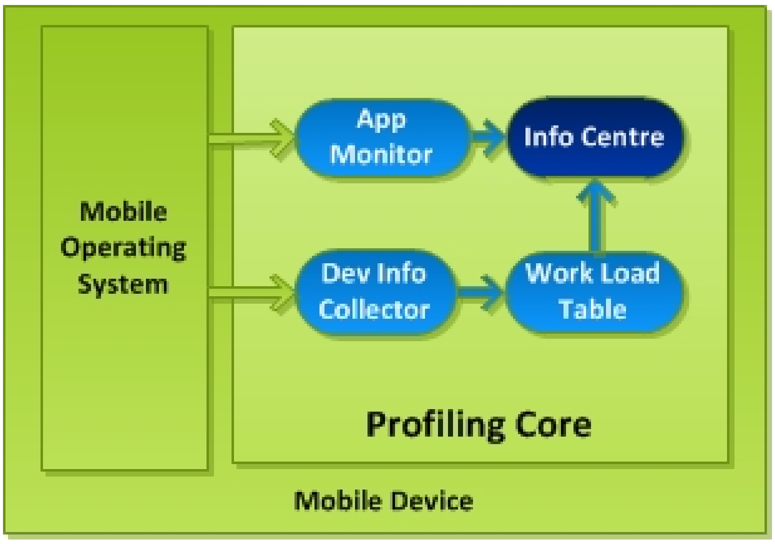

Figure 6.4: ASP Mechanism

ables pairs of devices with the same interest in certain content to retrieve part of that multimedia content from the content provider via a network interface (e.g. cellular) and exchange it against the missing part from its pair via another network interface (e.g. WiFi).

6.3.1 Energy-oriented Application-based System Profiling

The energy constraint information provided by ASP is used for quality adaptation and partner selection of content sharing. ASP builds an energy model of the mobile device, records power signature of each hardware subsystem, and calculates energy constraint score in real time. The block-level components of this algorithm along with the information exchange mechanism are illustrated in Fig.6.4. Once a new application is launched, *App Monitor* identifies it as the current running application. From the *Mobile Operating System* readings of CPU workload, wireless network card load, cellular interface utilisation, as well as the resolution of the display unit, battery characteristics, and battery energy level. *Dev Info Collector* is in charge with doing these readings in order to form the power signature of the application on this device. Based on this information, *Profiling Core* stores device features, maintains different application profiles, evaluates the delivery QoS level, and calculates the expected battery life. More details are presented in chapter ASP and AWERA.

6.3.2 Energy Efficient Quality Adaptation

A content quality adaptation algorithm is in place for energy-efficient multimedia content delivery. The device can request degradation of the content quality level provided to the pair it belongs to, in order to reduce network traffic and importantly save energy. This adaptation enables devices to react to the dynamically changing network environments. The quality adaptation component of EDCAM is performed by DEAS [148]. The block level illustration of the quality adaptation mechanism is illustrated in chapter 4. Fig.6.5 illustrates the architecture of EDCAM. The *EDCAM Core* works with *ASP Profiling Core* and *DEAS Client Component*. The *DEAS Client Component* works with *EDCAM Server Component* for adaptive multimedia delivery. *EDCAM Core* uses *Cellular Data Receiver* to retrive data from the server, and uses *WiFi NIC* to share its portion of multimedia content with its partner.

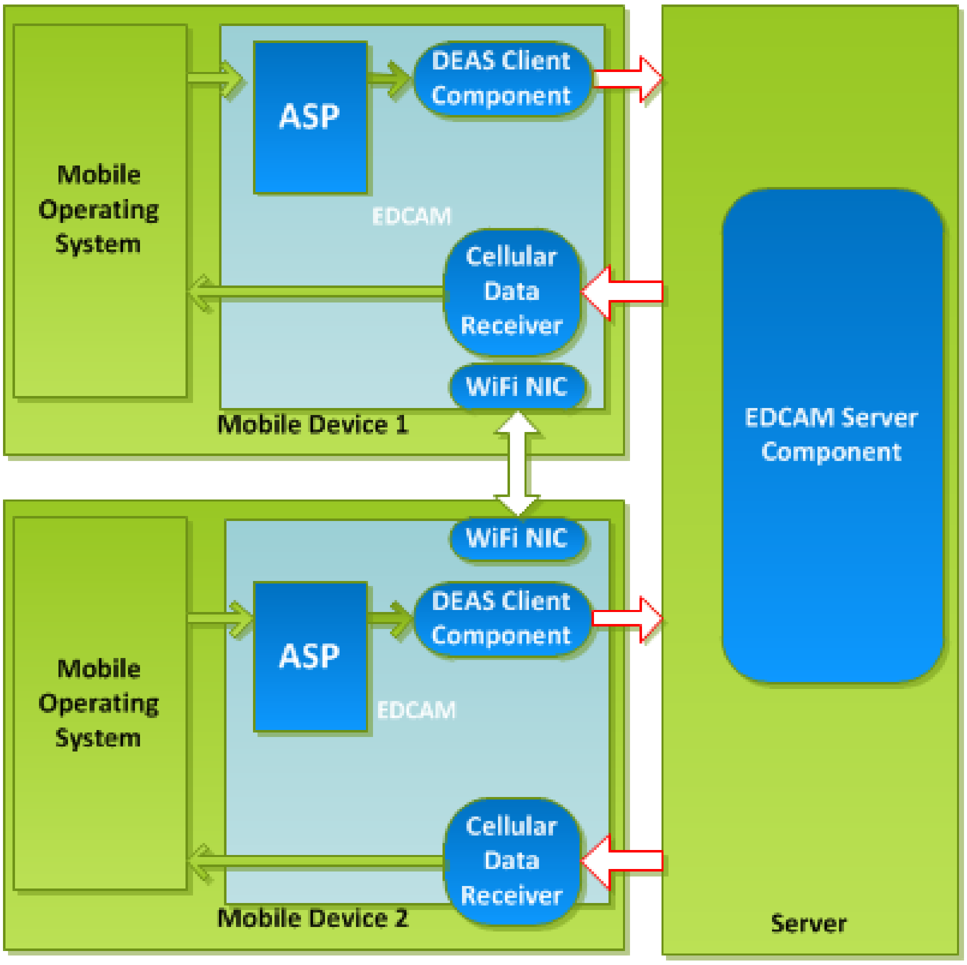

Figure 6.5: Architecture of EDCAM

6.3.3 Partner Selection in Cooperative Content Delivery

If multiple devices sharing the same interest (including quality level and context) are at present in the same neighbourhood, the interest level in content along with required quality level is used for the selection of eligible sharing partners. As cellular networks incur higher energy costs and have relatively lower bandwidth and often variable connectivity, this cooperative downloading approach is designed to conserve energy by encouraging the usage of the WiFi network interface for communication instead of the cellular interface. The energy saving will be analysed in the "Benefit Analysis" subsection and demonstrated in the test results section. Fig.6.6 shows the architecture of the sharing partner selection mechanism. *Profiling Core* provides "profile" data that

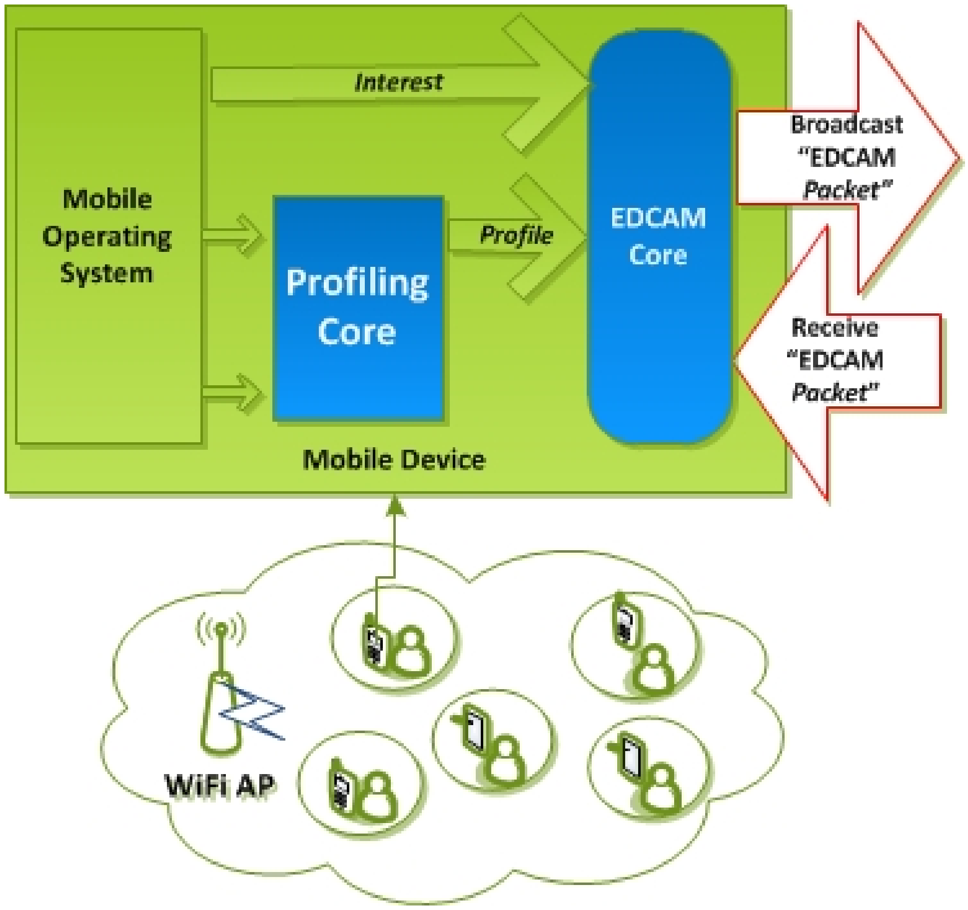

Figure 6.6: Sharing Partner Selection

states the current energy constraint of the device to the *EDCAM Core*. The current application on the operating system provides the "interest" level in the content to the *EDCAM Core*. *EDCAM Core* aggregates these two pieces of information in an "EDCAM Packet" and broadcasts it. Among all the devices that listen to these broadcasts from the sources of the packets with the same "interest", the one with the most critical energy constraint is selected as the content sharing partner. The partner selection and partnership establishment mechanism will be explained in more details in the following subsections.

6.4 EDCAM Algorithm

6.4.1 Energy-oriented Device System Profiling

Energy-oriented Application-based System Profiling (ASP) constructs a Profile that includes: *Energy Model* that takes workload of device components and calculates the corresponding battery discharge ; *Application Profile Table* that records application power signature, namely the typical work load on the hardware components for each application. These two elements are used to determine device energy constraints, which are used for quality adaptation and cooperative downloading.

6.4.2 Energy-efficient Quality Adaptation Algorithm

The Energy-efficient Quality Adaptation Algorithm requests the server to perform adjustments to the delivery rate according to two metrics: battery expected lifetime that is calculated based on the data collected from System Profiling; and perceived quality that is estimated using PSNR of the monitored active video application.

As revealed by extensive tests [169], the variation of traffic volume and decoding effort caused by different video quality levels has major impact on energy efficiency when compared with that of link quality, network load and transport protocol. Consequently Energy-efficient Quality Adaptation Algorithm focuses on the above two metrics. The details of the quality adaptation algorithm used in EDCAM are presented in the chapter of DEAS.

6.4.3 Two-party Cooperative Downloading

6.4.3.1 Motivation for Two-party Sharing

The cooperative downloading algorithm involves two parties in the sharing process, rather than multiple devices as proposed by other content sharing solutions [56, 60, 57, 58, 171]. The proposed design is backed by several facts.

The existing simple content sharing solutions assume the availability of WLAN, which is not realistic due to the limited deployment of free WiFi hot spot. Moreover the WiFi WLAN can be very congested, and consequently it is not suitable for multimedia delivery with high bandwidth demand and tight QoS restrictions.

Without existing WiFi AP, an ad-hoc wireless LAN with corresponding routing algorithm is needed for multiple nodes content sharing. The relatively larger overhead of group management brings challenges to the improvement of energy efficiency and system performance. The deployment of ad-hoc wireless LAN itself is against the goal of energy saving, this is because the connection setup and group management need peer-to-peer information exchanges between all the parties. Once any device encounters difficulties in maintaining connectivity, the whole connection setup process that introduces all the overhead has to be invoked all over again.

In addition, the download scheduling among devices is challenging and complex. Any distributed complex calculation will affect negatively the individual smart device, which is again, against energy saving principle. In terms of performance, the involvement of multiple parties pose challenges to the scheduling of downloading. Facing this challenge, a bigger buffer with intelligent buffer management algorithm is needed in the gateway device. Even if a dedicated gateway is deployed, instead of electing one device to act as a gateway, this approach introduces delay and difficulties in stitching partial content together.

Last, it is unlikely that a group of heterogeneous devices simultaneously demands the same quality level as there optimal choice. Hence re-clustering happens regularly to avail group quality adaptation. In contrast, it's a feasible and agile design that a device performing quality adaptation shares content with another individual device demanding for the same quality level.

In conclusion, regarding energy efficiency as the design philosophy, a two party content sharing is chosen for ease of deployment and less overhead in the sense of both connection management and download scheduling.

6.4.3.2 Content Sharing Mechanism

In the restricted space where people are densely distributed with relatively stable mobility pattern, for example, convention centres, sports stadiums and high speed trains, it is highly likely there are more than one device showing common interest in such scenario. The Content Sharing mechanism is described in Algorithm 2.

Algorithm 2 Partnership Establishment and Content Sharing Mechanism

```
Device Needs to Request Content from the Server:
Calculate(Quality, Exp_Life);
Broadcast(Quality, Content);
while True do
    Wait(Grace_Period);
    if Exist(Reply) then
        Find(Minimal_Exp_Life, Replies);
        Communicate(Partner);
        Connect(Server, Quality);
        Sharing_and_Adaptive_Delivery
    else if NonExist(Reply) then
        Connect(Server, Quality);
        Adaptive_Delivery
    end if
    Wait(Time_Out);
    Calculate(Quality, Exp_Life);
    Broadcast(Quality, Content);
end while
```

When a device's energy constraint is tight, it can initiate a search of partner for content sharing-assisted adaptive multimedia delivery. Before downloading any content, a device will use the WiFi interface to broadcast its interests of content (the context and the corresponding quality level). In the EDCAM software implementation, all the EDCAM enabled devices will use the same subnet and the interest information is sent to the broadcast IP address of that subnet. While DHCP is unavailable, the IP assignment happens in the cloud. A grace period is assigned to allow the device wait for any nearby device sharing the same interest to join and download together. Meanwhile the nearby devices demanding ”the same content” of ”the same quality level” will response the request with a control packet of its energy constraint information as in (5.1). If no device is joining by the end of this period, the current device will proceed to perform energy-efficient quality adaptive downloading alone.

Facing multiple replies as in Fig.6.7, the initiating device will establish a partnership with the most energy constraint device based on the energy constraint information retrieved from the received response packets. This design aims to maximise the benefit. When quality adaptation is needed, the device will first communicate with its partner for permission. If permission is granted, they will both request for quality adaptation at the same level. If they cannot come to an agreement because the QoS or expected life time exceed the partner’s threshold, the device will search for another sharing partner or, if not available, will transfer the content alone. The criteria for quality adaptation has been discussed in the previous section.

Once a partner has been selected, a TCP-like three way handshake procedure is used to establish a WiFI connection with the sharing partner. The handshake procedure is demonstrated in Fig.6.7. This is implemented as an extended version of RTSP: RTSP-E. For RTSP-E, devices may broadcast *Interest* packet enclosing *interest of content* and wait for *Response* packets enclosing *energy constraint* and *expected quality levels*. Once a partner is selected, the initiating device may start to establish partnership. For example, in Fig.6.7, device A sends *SYN-ACK* control packet to the selected Device B to confirm its willingness to corporately download. *SYN-ACK* control packet includes: *SeqNo* - the sequence number for device B to start with, and *LEN* - the length of each sharing unit ’LEN’. In this case, A will download from SeqNo+LEN, B will download from SeqNo. Once B receives *SYN-ACK* packet, it will send *ACK* control packet to A to establish the connection. Next, they will both download their part of length ’LEN’ and exchange as soon as they have it.

Once the bound is established, both devices use cellular interface to download the

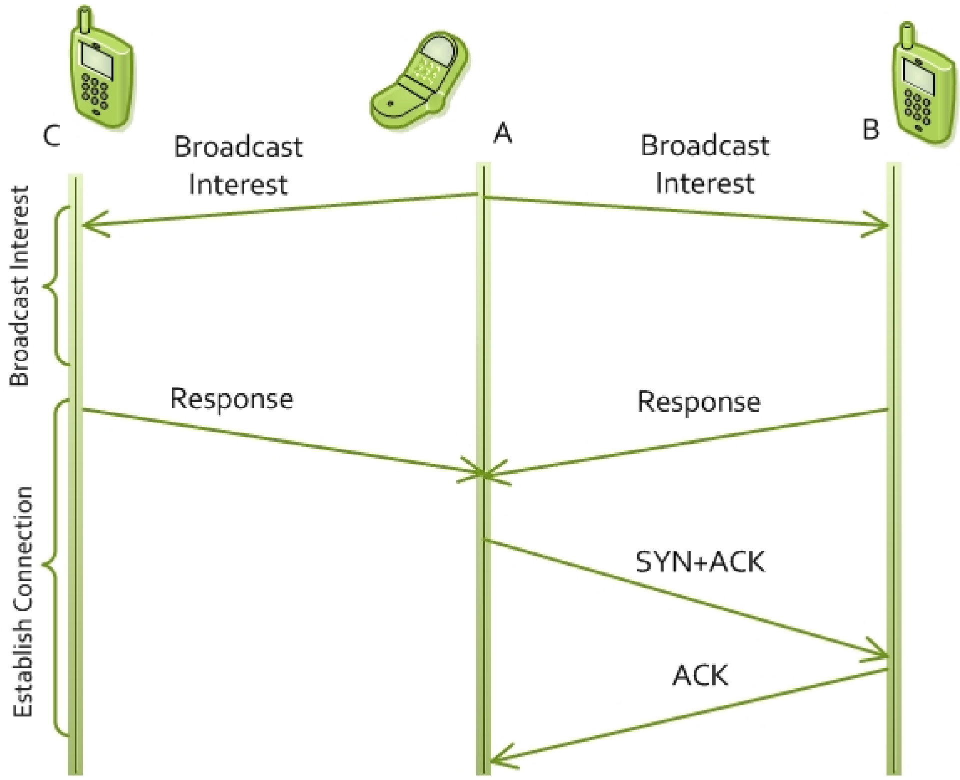

Figure 6.7: Handshake Procedure to Establish a Connection with the Partner for Content Sharing

content corporately, and use their WiFi interface to share the content between each other.

6.4.4 Overhead and Benefit Analysis

This subsection presents the energy benefit analysis when employing two party cooperative downloading and quality adaptation over WiFi and cellular networks, respectively. Table 6.1 presents the definition of parameters used in this discussion.

$$t_{Tra} = \frac{\alpha_{\text{File}}}{\beta_c} \tag{6.1}$$

Table 6.1: Parameter Definitions

Variable	Definition
α_{File}	Multimedia file size (bit)
α_{o}	The amount of overhead (bits)
β_{c}	Cellular network interface bandwidth (bps)
β_{w}	WiFi interface bandwidth (bps)
γ_{c}	The size of cellular traffic (bits)
γ_{w}	The size of WiFi traffic received and transmitted (bits)
ϵ_{c}	The price of transmit 1 bit over cellular network (euro)
ϵ_{w}	The price of transmit 1 bit over Wifi network (euro)
P_{c}	The power of Cellular interface (W)
P_{w}	The power of WiFi interface (W)
E_{Total}	Energy consumption of device using proposed algorithm (Ws)
E_{Tra}	Energy consumption of device using only cellular interface (Ws)
$\text{E}_{\text{Benefit}}$	Energy efficiency benefit (Ws)
$\text{M}_{\text{Benefit}}$	Monetary benefit (euro)

$$E_{Tra} = P_c \cdot t_{Tra} \tag{6.2}$$

Equation (6.1) represents the transmission time of the required multimedia content via the cellular interface using the traditional approach, where α_{File} is the size of the multimedia file and β_{c} is cellular network interface bandwidth. Consequently, equation (6.2) shows the required energy for successful content transmission using the cellular network only.

$$\gamma_c = \alpha_{\text{File}} \cdot \theta \tag{6.3}$$

$$t_c = \frac{\gamma_c}{\beta_c} \tag{6.4}$$

$$\gamma_w = \alpha_o + (1-\theta)\alpha_{\text{File}} \tag{6.5}$$

$$t_w = \frac{\gamma_w}{\beta_w} \tag{6.6}$$

$$E_{Total} = P_c \cdot t_c + P_w \cdot t_w \tag{6.7}$$

Equation (6.3) calculates the size of the cellular traffic if the cellular interface exchanges only a portion θ ($0 \geq \theta \leq 1$) of the original multimedia content. Hence the data transmission time over cellular interface is shown in (6.4). Equation (6.5) computes the size of the WiFi traffic including the control overhead, the portion of file received from the sharing partner and the portion of file it sends to the sharing partner. Since the RTSP header length is 12 bytes and the connection establishment happens at the very beginning only, α_{o} is very small in comparison with α_{File}, which easily exceeds tens of hundreds of MB. In addition, the WiFi interface exchanges overhead control traffic associated with EDCAM. Equation (6.6) shows the data transmission time over WiFi interface.

Since EDCAM uses automatic two-party cooperative download, θ = 0.5 for the final equation.

Based on this assumption, as long as cellular interface's power-per-bit value is higher than the WiFi interface's, $\mathrm{E_{Total}}$ from equation (6.7) will be less than $\mathrm{E_{Tra}}$ from equation (6.2). This is usually true, for example, the cellular interface consumes 18 times energy than the WiFi interface on ipad [56]. Consequently, $\mathrm{E_{Benefit}}$ from equation (6.8) is positive.

$$E_{Benefit} \quad = \quad E_{Tra} - E_{Total} \tag{6.8}$$

In addition, energy-efficient quality adaptation algorithm can further reduce the size of data to deliver, while maintaining the QoS level to satisfactory. According to our theoretical analysis, the proposed hybrid solution is able to conserve energy for smart mobile devices while receiving multimedia content.

$$M_{Benefit} \quad = \quad \epsilon_{\mathrm{c}} \cdot \theta\alpha_{\mathrm{File}} + \epsilon_{\mathrm{w}} \cdot (1-\theta)\alpha_{\mathrm{File}} \tag{6.9}$$

The monetary benefit $\mathrm{M_{Benefit}}$ is demonstrated by equation (6.9), where ϵ_c symblises the price of transmitting 1 bit of data over the cellular networks and ϵ_w symblises the price of transmitting 1 bit of data over the Wifi networks. Since the content α_{File} that is to be transmitted over cellular network is reduced to θ (up to half when θ=0.5), the expenses $\mathrm{M_{Benefit}}$ are reduced up to half for cellular network transmission when θ = 0.5. Meanwhile, the user will pay for Wifi data for the portion of (1-θ). The cost of Wifi data transmission is considered inexpensive if not free. Notably, the handshake happens in WiFi interface, which is often free of charge.

6.5 Simulation Testing and Result Analysis - EDCAM

6.5.1 EDCAM NS-3 Model and Simulation Testing Environment

EDCAM was developed, simulated and tested in Networks Simulator 3 [1]. Heterogeneous networking simulation scenarios were built to test the proposed solution and compare EDCAM with other solutions.

The NS-3 model of EDCAM is based on the implementation of DEAS. However, the sharing mechanism between two parties via WiFi has to be implemented. The previously presented ”two party sharing mechanism” in chapter 5 is implemented in NS-3 Model. Additionally an LTE energy model which was not available by default had to be implemented. Fig.6.8 presents the class diagram of EDCAM implementation in NS-3. All the major classes are included in the diagram.

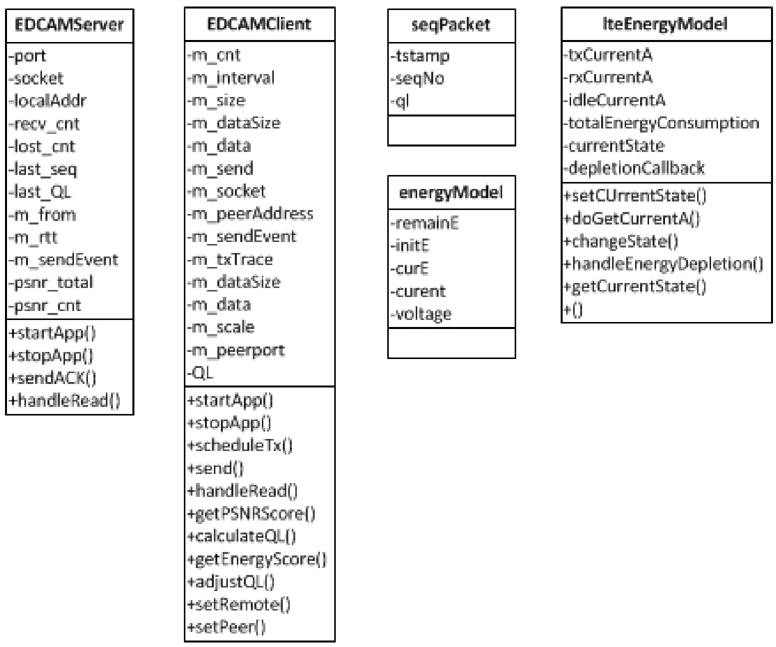

Figure 6.8: Class Diagram of EDCAM Implementation in NS-3

The LTE energy model class ”LTEenergyModel” is implemented to link to the LTE NIC. LTE NIC will notify ”LTEenergyModel” when its state changes (”idle” ”send”

[1]Network Simulator 3, http://www.nsnam.org/

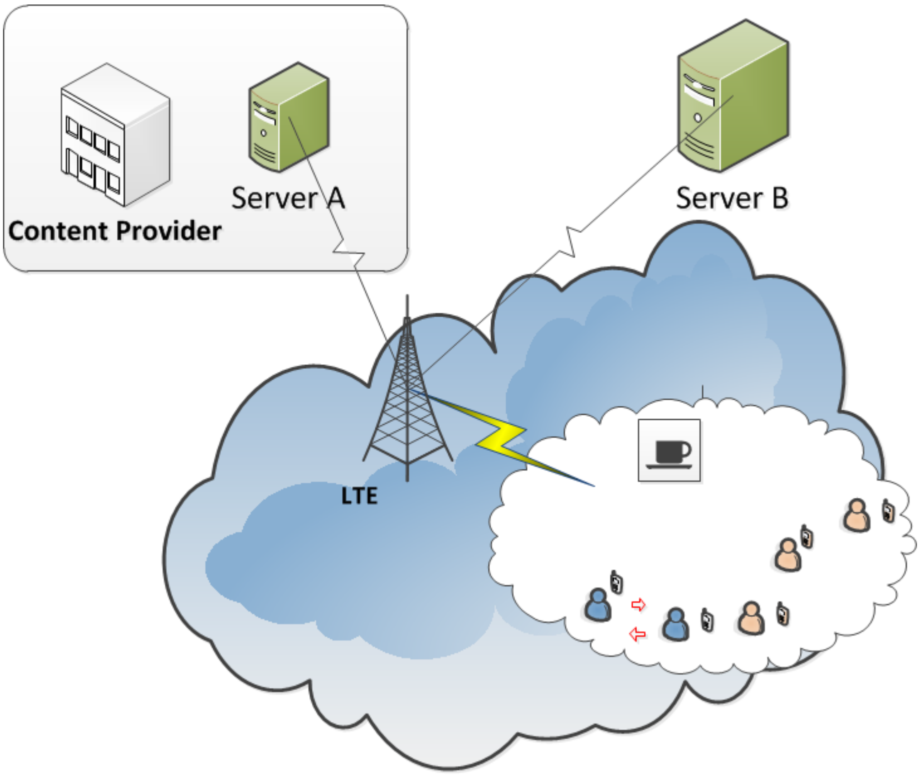

Figure 6.9: Simulation Topology - EDCAM

”receive” ”control”). When the sate changes, ”LTEenergyModel” will deduct the corresponding energy from the energy source attached to the device. The cross layer two party content sharing presented in chapter 5 is also implemented in ”edcam_app” class. Once data packets are received from LTE NIC, the ”edcam_app” will deliver it to upper layer and send a copy to its sharing partner via WiFi NIC. Each device caches and synchronises the content received from both interfaces before playing out. The DEAS components of EDCAM has been explained in chapter 7 already.

6.5.2 Simulation Scenario - Heterogeneous Wireless Network Environment: LTE and WiFi

The simulation model was developed in Network Simulator 3 [2] to validate the proposed solution for energy saving at the client devices. The topology deployed in the

[2]Network Simulator 3, http://www.nsnam.org/

simulation is illustrated in Fig.6.9. LTE cellular network and fast wired network comprise a heterogeneous networking environment. In Fig.6.9, over the WiFi network, the two Wifi users in dark colour exchange the portion of content they downloaded from content provider.

A group of 10 mobile devices are evenly deployed in 100 square meter area. All the devices are equipped with both a WiFi interface and a LTE interface. A LTE base station is deployed 1000 meters away in west direction to this area. The content provider delivers video content via the LTE base station to three LTE subscriber devices. Server B is transmitting background traffic to the other devices via LTE base station to load the LTE network.

Mobility is implemented in this simulation. As mentioned before, the cooperative multimedia delivery is often needed in trains, ships, cafe shops, convention centres, etc. Consequently, mobility considers the user devices randomly "walk" around in the deployed area at the speed of 0.5m per second during the simulations. The simulation runs for 15 times.

6.5.2.1 Simulation Settings

The video content is of H.264/MPEG-4 format and stored at 5 quality levels with the average bit rates of 1920 kbps, 960 kbps, 480 kbps, 240 kbps and 120 kbps, respectively, listed from the highest to the lowest quality levels. Three devices receive the above multimedia content.

The number of subscribers of background traffic differs in the six scenarios considered in the simulations as follows: A) 2 devices; B) 3 devices; C) 4 devices ; D) 5 devices; E) 6 devices; F) 7 devices. This is summarised in Table6.2. For each flow, the tests were conducted with background traffic bit rate being set as 1Mbps and 2Mbps respectively. The growth of subscriber number yields the growth of number of flows participating LTE transmission in the cell. This is to find out how the compared solutions cope with the increasingly congested network environment.

Table 6.2: Background Traffic Classes: Number of Background Traffic Subscribers for EDCAM Scenario

Background Traffic Class	Number of Subscriber
A	2 Devices
B	3 Devices
C	4 Devices
D	5 Devices
E	6 Devices
F	7 Devices

Since there were 10 devices in total with 3 adaptive video content subscriber, the maximum number of flows is 7. From the simulation, the results from 1 background traffic flow and 2 background traffic flows are identical. Hence it is trivial to show the 1 device background traffic subscriber scenario.

The simulation configuration limited the resources allocated to the cell, and the networks are overloaded in the above scenarios. This setting fully explores how the solutions behave differently facing different network conditions. The results obtained with different background traffic data rates are recorded separately for clear comparison.

In order to perform realistic modelling and simulations, the power settings are configured according to comprehensive measurements with a Google Nexus One mobile device in a real test bed [169]. The power readings when playing video content at different quality levels are recorded. The power settings for local playback of the content are 1157 mW (1920 kpbs), 811 mW (960 kbps), 673 mW (480 kbps), 612 mW (240 kbps), 560 mW (120 kbps) for the five quality levels considered..These power settings on WiFi interface for transmitting are 288 mW (1920 kpbs), 211 mW (960 kbps), 168 mW (480 kbps), 152 mW (240 kbps), 140 mW (120 kbps) for the five quality levels considered. The power settings on LTE interface are moderately set as 4 times higher than WiFi interface [172]: 1445 mW (1920 kpbs), 1022 mW (960 kbps), 841 mW (480 kbps), 764 mW (240 kbps), 699 mW (120 kbps). The initial battery capacity is 180 J.

The simulation duration is 100 s. These are both reduced from real life values for the simulation while the validity remains unaffected. The above settings are summarised in Table6.3.

Table 6.3: Simulation Settings for EDCAM

Parameter	Value
Wired Link Bandwidth	100 Mbps
Wired Link Latency	2 ms
User Mobility Speed	0.5mps
Battery Voltage	3.7 V
Battery Capacity	186.48J
Thread of Battery Life	100 Sec
Thread of PSNR	30 dB
Bit Rate QL1	1920 kbps
Bit Rate QL2	960 kbps
Bit Rate QL3	480 kbps
Bit Rate QL4	240 kbps
Bit Rate QL5	120 kbps
Background Traffic Bit Rate	2 Mbps
Device Power (QL1)	1445 mW
Device Power (QL2)	1022 mW
Device Power (QL3)	841 mW
Device Power (QL4)	764 mW
Device Power (QL5)	699 mW
Simulation Duration	100 Sec

EDCAM is compared against Non-adaptive, DEAS and ESTREL. ESTREL uses the similar energy saving mechanism of adjusting streaming bit rate. Facilitated by scalable video coding scheme, ESTREL unsubscribes enhancement layer data traffic when energy level is low in order to conserve energy. ESTREL is configured as in [145]. DEAS takes energy metric and quality metric to best balance the quality and energy efficiency. It follows the description from chapter 4. In order to improve energy efficiency to more extent, EDCAM improves DEAS by introducing cooperative mechanism that allows two devices sharing content.

6.5.2.2 Simulation Results

The above delivery schemes are compared in terms of average values of delay, jitter and PSNR in order to illustrate their levels of quality of service and estimated user perceived quality measured using PSNR. The performance of the solutions is assessed in terms of the device energy consumption. The results for EDCAM, DEAS, ESTREL and Non adaptive (NonAd) are shown in Table 6.4, Table 6.5, Table 6.6 and Table 6.7 respectively.

Table 6.4: EDCAM Scenario: Simulation Results for EDCAM

Background Traffic Class	A 1Mbps	A 2Mbps	B 1Mbps	B 2Mbps	C 1Mbps	C 2Mbps	D 1Mbps	D 2Mbps	E 1Mbps	E 2Mbps	F 1Mbps	F 2Mbps
Energy Consumption (J)	84.48	92.70	86.29	92.72	89.33	94.33	92.06	95.28	92.12	95.29	91.64	94.33
Delay (Sec)	0.0027	0.0027	0.0030	0.0028	0.0027	0.0028	0.0028	0.0029	0.0030	0.0029	0.0031	0.0032
Jitter (Sec)	2.030	2.072	2.034	1.176	2.031	1.172	3.083	4.049	3.145	2.180	3.052	3.500
PSNR (dB)	38.54	36.67	38.54	36.67	38.53	36.67	37.90	36.67	36.66	36.67	36.67	36.67

Table 6.5: EDCAM Scenario: Simulation Results for DEAS

Background Traffic Class	A 1Mbps	A 2Mbps	B 1Mbps	B 2Mbps	C 1Mbps	C 2Mbps	D 1Mbps	D 2Mbps	E 1Mbps	E 2Mbps	F 1Mbps	F 2Mbps
Energy Consumption (J)	107.44	114.78	111.85	116.78	113.11	116.69	113.65	116.69	114.28	116.69	115.39	112.89
Delay (Sec)	0.0050	1.0030	0.0060	1.0100	0.0050	1.0120	0.0610	3.0500	0.0950	4.0190	2.0590	6.1410
Jitter (Sec)	3.020	5.095	3.040	5.070	3.033	5.079	5.040	7.080	6.120	8.060	8.116	9.168
PSNR (dB)	37.30	29.43	37.32	29.40	35.44	29.41	30.21	26.19	29.69	28.44	26.45	28.18

Table 6.6: EDCAM Scenario: Simulation Results for ESTREL

Background Traffic Class	A 1Mbps	A 2Mbps	B 1Mbps	B 2Mbps	C 1Mbps	C 2Mbps	D 1Mbps	D 2Mbps	E 1Mbps	E 2Mbps	F 1Mbps	F 2Mbps
Energy Consumption (J)	129.34	133.73	131.36	133.73	131.77	133.73	133.05	133.73	133.73	133.81	133.73	133.73
Delay (Sec)	0.0058	6.0792	0.0060	6.0600	0.0059	6.0792	2.0760	5.0373	4.0470	5.0420	5.0460	5.0304
Jitter (Sec)	6.060	4.058	6.180	3.180	6.038	4.058	6.013	5.037	9.060	11.120	11.114	12.115
PSNR (dB)	52.10	22.19	24.67	24.57	24.19	22.19	24.20	14.14	23.10	9.67	18.84	7.22

Table 6.7: EDCAM Scenario: Simulation Results for NonAd

Background Traffic Class	A 1Mbps	A 2Mbps	B 1Mbps	B 2Mbps	C 1Mbps	C 2Mbps	D 1Mbps	D 2Mbps	E 1Mbps	E 2Mbps	F 1Mbps	F 2Mbps
Energy Consumption (J)	135.86	155.61	137.00	155.62	155.61	145.75	155.61	155.61	155.61	155.52	155.62	155.61
Delay (Sec)	0.0087	5.0408	0.008	5.0400	5.0444	5.0400	5.0442	5.0482	5.1200	5.0300	5.0513	4.1238
Jitter (Sec)	9.129	2.106	9.120	2.130	2.134	2.130	14.097	5.100	12.130	6.080	13.164	11.091
PSNR (dB)	24.58	10.08	24.57	10.09	10.10	22.17	13.87	7.89	11.77	6.50	8.67	5.59

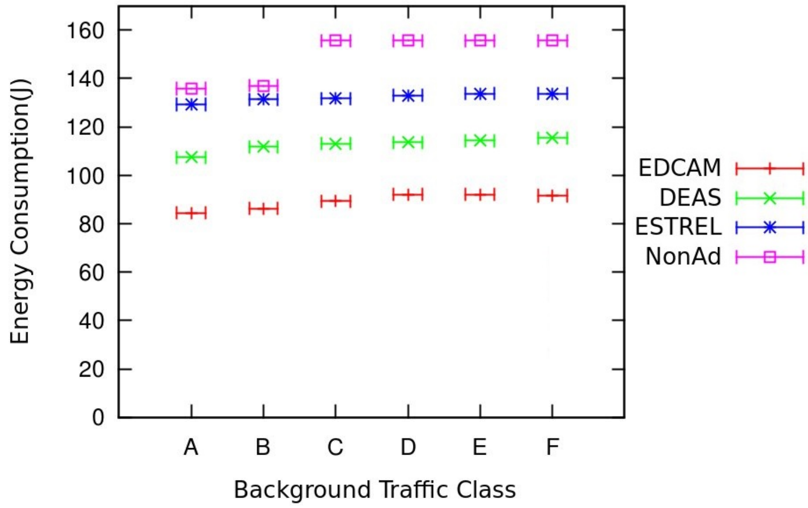

Figure 6.10: EDCAM Scenario: Remaining Energy - 1Mbps Background Traffic

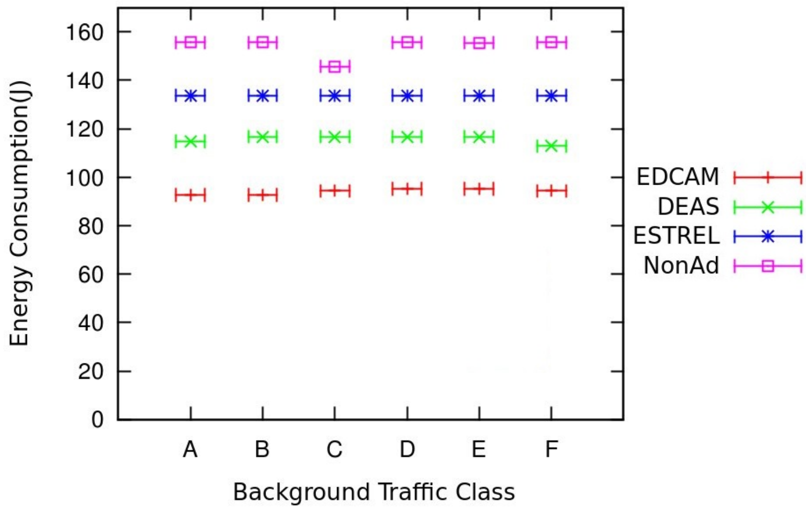

Figure 6.11: EDCAM Scenario: Remaining Energy - 2Mbps Background Traffic

Three key observations from the simulation results are as follows: adaptive solutions (ESTREL, DEAS, EDCAM) successfully conserve energy by reducing the traffic volume to be transmitted; QoS (DEAS, EDCAM) oriented adaptation is effective in improving user perceived quality; cooperative solution (EDCAM) achieves further energy saving as the LTE interface is less often involved in data transmission.

Fig. 6.10 and Fig. 6.11 clearly show that EDCAM demonstrates substantial energy saving. It used just over half of the energy than NonAd. Since ESTREL starts to adapt

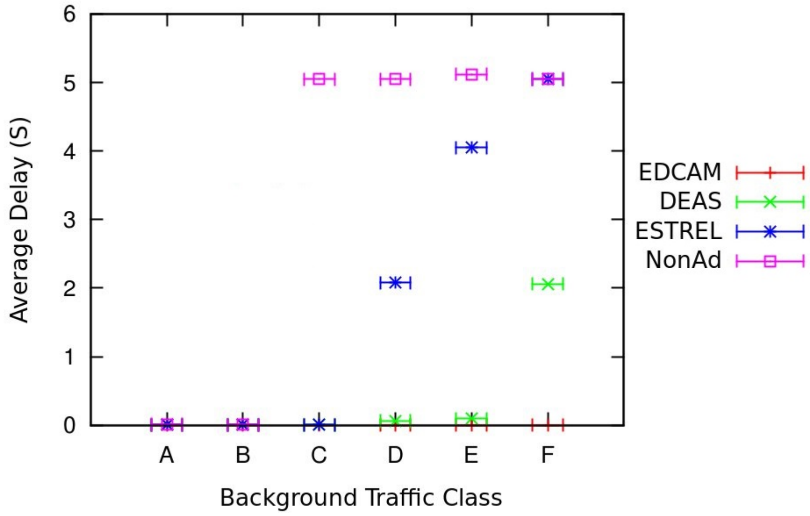

Figure 6.12: EDCAM Scenario: Average Delay - 1Mbps Background Traffic

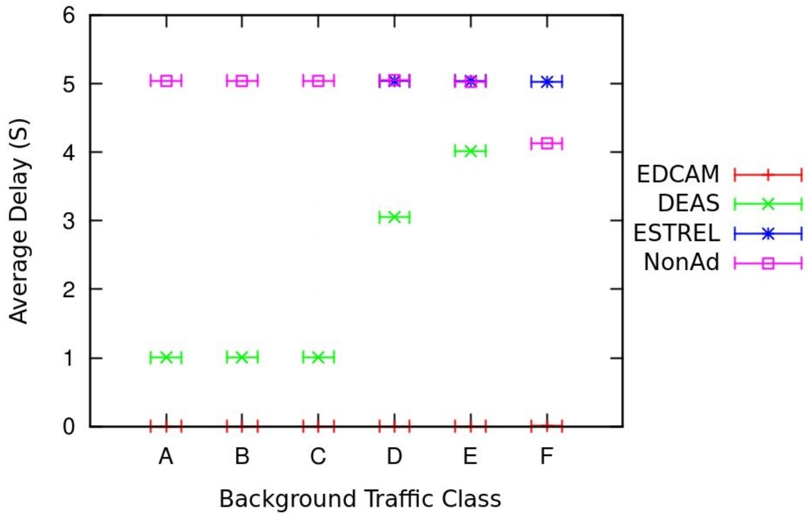

Figure 6.13: EDCAM Scenario: Average Delay - 2Mbps Background Traffic

traffic bit rate only when energy level is already low, the saving by ESTREL is not as high as that of DEAS and EDCAM. By introducing the WiFi interface-based data sharing, EDCAM improves the performance of DEAS by 20 percent. Therefore the cellular interface usage reduction is very useful in saving energy.

Since the recorded energy consumption is comprised of both local processing energy consumption and network transmission energy consumption, the energy saving is not proportional to the reduction of networking traffic. Still network traffic reduction

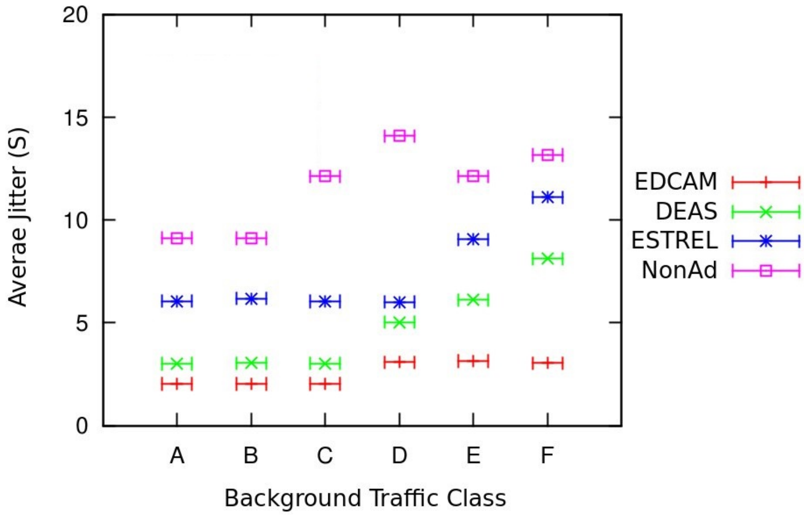

Figure 6.14: EDCAM Scenario: Average Jitter - 1Mbps Background Traffic

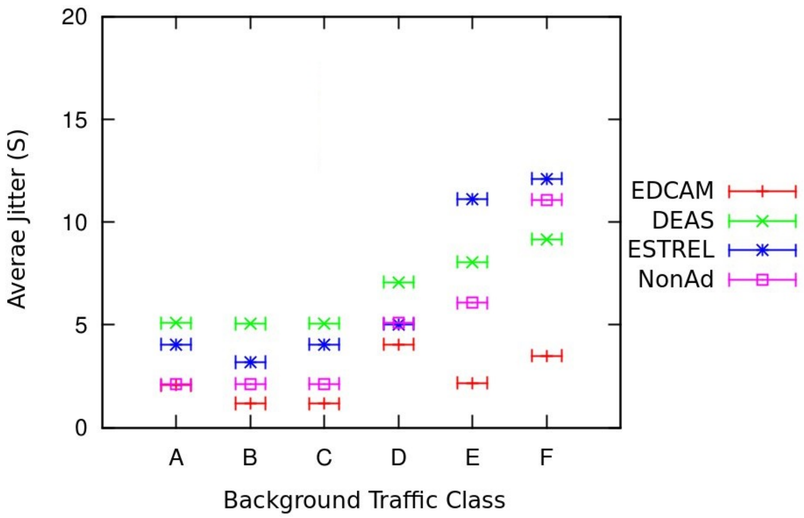

Figure 6.15: EDCAM Scenario: Average Jitter - 2Mbps Background Traffic

results in less local processing. In addition, despite the differences of energy efficiency between multiple network interfaces [172], the cooperative approach is not able to ease the decoding and display effort, namely the energy consumption of the local playback.

The increased background traffic volume does not affect the energy consumption much. This is because the traffic volume to the end user remains similar for adaptive solutions when the network is fully loaded. On the contrary, the energy consumption of NonAd increased facing heavy traffic. However, the increased background traffic

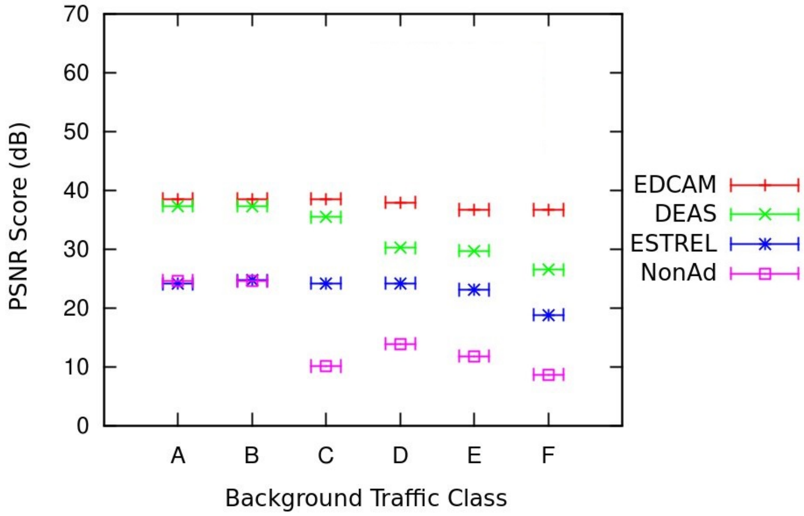

Figure 6.16: EDCAM Scenario: Average PSNR - 1Mbps Background Traffic

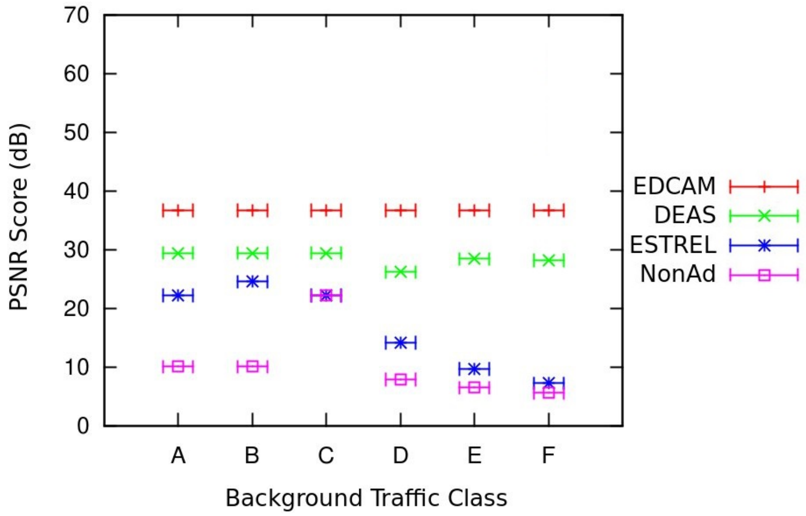

Figure 6.17: EDCAM Scenario: Average PSNR - 2Mbps Background Traffic

volume brings changes to the user perceived quality as in the following discussion.

Adaptively reducing traffic results in better performance. Reducing cellular interface usage boosts also the performance due to the WiFi interface offload for part of the multimedia data. Fig. 6.12 and Fig. 6.13 present Non-adaptive delivery had significantly higher delay than adaptive solutions. Additionally, EDCAM shows more consistent performance in all the scenarios. With 1Mbps background traffic, ESTREL started to struggle in scenario D. With 2Mbps background traffic, ESTREL can not delivery from scenario D onwards.This is because ESTREL did not adapt until energy is low. Therefore the total performance is inferior to that of DEAS and EDCAM. Although DEAS is designed to balance QoS and energy efficiency, DEAS failed to maintain low delay starts from scenario E facing 1Mbps background traffic. Given 2Mbps background traffic, DEAS struggled even earlier. However, with half of the traffic shared in WiFi interface, EDCAM easily shows stable and very low delay through at all the scenarios.

While high delay values are annoying for multimedia content receive. They can be accumulated using buffering. However, high jitter values affect user perceived quality. In Fig.6.14, it can be noted how EDCAM is superior to the other solutions while adaptive solutions are much better than NonAd. Even when facing heavy background traffic as in Fig.6.15, EDCAM demonstrated stable and very good performance. DEAS suffered from relatively high jitter, which can be introduced by frequent adaptations. However, the jitter value from DEAS is more stable when heavier traffic was introduced as in scenario E and F.

Fig.6.16 and Fig.6.17 show how the PSNR scores confirm EDCAM is superior to the other solutions in terms of QoS performance. First, the scores are stable in all the scenarios for EDCAM. DEAS has better results in both 2Mbps and 1Mbps background traffic situations. The adaptive solutions managed to achieve above ”good” levels in all the scenarios, while EDCAM and DEAS have scored ”excellent”. In Fig.6.17, it can be seen how ESTREL degraded its quality to ”acceptable” and ”poor” levels in heavy traffic conditions. As expected, NonAd performed poorly except in very light background traffic conditions as in scenario A and B as shown in Fig.6.16.

6.6 Chapter Summary

This chapter presents EDCAM - a novel hybrid solution for content sharing and quality adaptation that aims at energy-efficient multimedia content delivery to smart devices. EDCAM uses ASP for device energy constraints calculation, it combines energy saving two-party cooperative downloading scheme which reduces group management overhead and utilises the multi-home capability of smart devices, and an energy efficient quality adaptation algorithm that adapts video quality for energy saving. EDCAM saves energy while maintaining high QoS in heterogeneous wireless networking environment. Extensive tests have demonstrated how EDCAM outperforms three state of the art solutions resulting in significant improvements in terms of both energy efficiency and user perceived quality levels.

Chapter 7

Conclusions and Future Work

7.1 Overview

The latest interest in multimedia delivery to diverse mobile devices is increasing significantly. Apart from the focus on quality, lately there is increasing interest in energy efficiency. To deliver multimedia content efficiently with respect to both performance and energy consumption is a major challenge. Adaptive solutions are seen as a solution to this challenge.

In this context Martin et al [5] have indicated that there is a need for a more sophisticated adaptive approach that takes the nature of the device, the current application and current battery level, analyses and processes the data and adjusts the delivery process in an innovative manner in order to prolong the battery life for mobile devices. This is a little researched area in the energy consumption space.

Filling this gap, this thesis first presents a survey on the state of the art of hardware and software energy saving solutions for network content delivery. This has contributed to a comprehensive survey paper on the this subject. It suggests that devices have diverse energy characteristics (software-wise and hardware-wise) which need to be fully explored before proposing energy saving solutions for mobile devices.

Next this thesis has proposed novel solutions for energy efficiency as follows: *ASP*, an energy-oriented application-based system profiling for mobile devices. This is a self-constructive adaptive regression based energy modelling software solution. *AW-ERA*, device differentiated energy efficient adaptive routing algorithm. *DEAS*, device differentiated energy efficient adaptive multimedia delivery solution. *EDCAM*, device differentiated energy efficient cooperative adaptive multimedia delivery solution. Next these contributions are summarised in details.

7.2 Contributions

This thesis presents the research outcome on device differentiated energy efficient routing algorithm and multimedia content delivery solution in heterogeneous wireless network environment. The contributions consist of the following:

Energy-oriented Application-based System Profiling for Mobile Devices - ASP and The Prototype System on Android Platform

ASP includes features from several existing models. As Sesame, battery interface is used instead of external hardware in the measurement of devices' energy depletion process. This enables self-constructive software-based device independent energy modelling. Two benefits are achieved: practicality and device independence. For incremental model update, the training only happens when the device is charging to minimise the energy consumption on modelling. As PowerBooter, the model is trained using many variations of loads rather than some typical loads for several existing applications. This effort is made in order to obtain a generalised model that suits many upcoming applications. Next are the major contributions of ASP.

- Application power signature is proposed to reduce monitoring overhead. This is because individual application shows distinctive energy characteristics on subsystems. This effort reduces the frequency of obtaining system parameters, including module usage data and the real time value of battery's current and voltage , in the monitoring process.

- The proposed model does not rely on external meters to validate the model. This is because external meters is not available in daily device usage, and is most likely not to be used for model updating and further training during daily device use.

- The energy modelling technique adopts incremental updating in order to adapt the model to the change of battery worn and user preferences.

- Field tests demonstrate the difference between LCD and OLED in terms of energy characteristics. Many works have been done on OLED. However a realistic yet different energy model for LCD was derived, which is often used for big screens, particularly tablet screens.

- A better refined CPU energy model was derived and problems caused by multi-core and multi-frequency of contemperory CPUs were addressed. These problems have not been fully addressed in the existing works.

- Many papers explored the construction of a power model for mobile-phones, but to our best knowledge, the proposed model is the first for tablet devices. Such a model is important since it permits us to analyse which component the energy is being spent on of the tablet. The model provides energy constraint information with higher rate that the battery interface. Moreover, it empowers the devices with the capability of energy constraint estimation.

Device Differentiated Energy Efficient Adaptive Routing Algorithm - AWERA

AWERA is different from other existing proposals since it uses ASP for energy-oriented application-based system profiling, which manages context information for device differentiation and energy saving. According to the latest research [8] [9], the change of energy constraints caused by the context variation differs significantly for different application types. Besides, networking traffic processing is one of the most energy consuming tasks for the mobile devices. Based on these observations, AWERA is proposed to record the typical context of mobile usage for different applications and maintains such information in the form of application profile. The profile is updated along the lifetime of the mobile device usage. This approach reduces the overhead of monitoring and enables a learning mechanism so that the device can adapt itself to the context environment automatically. An initialisation phase is conducted once only for each individual device in order to make the learning process device independent. Moreover, the periodical updating mechanism makes it possible to keep information up to date following any change in the context.

Based on the obtained model and context information (energy constraint and network condition), an adaptive routing protocol is specially designed for energy efficient wireless routing. First, the context information is passed on to the network layer. In addition, link quality is also considered in the routing process. Consequently, a dedicated routing information processing center converts the context information and link layer feedback into a new metric, which is maintained in a special routing table to accommodate such energy constraint-related context information. The routing table will be periodically updated to suit the change of the context. Such cross layer effort makes AWERA effective in adapting to the change of context and efficient in energy consumption. With the above information, AWERA intelligently and adaptively makes routing decisions for networking traffic so that the energy efficient routes are adopted during wireless data transmission.

Testing of AWERA has shown how AWERA achieved better energy efficiency than two other state-of-the-art schemes with much lower delay and very little degradation in throughput. The testing results were obtained from three different sets of test. Three test beds were developed and different stage of the algorithm were tested. These tests involve different traffic type, topology, mobility, and clearly show the benefits of adopting AWERA in comparison with the other routing solutions.

Device Differentiated Energy Efficient Adaptive Multimedia Delivery - DEAS

DEAS is an adaptive energy efficient multimedia delivery solution designed to suit the needs of multimedia content delivery in heterogeneous network environment. The limited battery capacity of current mobile devices and increasing amount of rich media content delivered over wireless networks have driven the latest research on energy efficient content delivery over wireless networks. Many energy-aware research solutions have been proposed involving traffic shaping, content adaptation, content sharing, etc. The existing solutions focus on the delivery application without considering application running environment and device features that pose different energy constraints on the whole content delivery process. This paper presents *a Device characteristics-based differentiated Energy-efficient Adaptive Solution* (DEAS) for video delivery over heterogeneous wireless networks. DEAS constructs an energy-oriented system profile including power signatures of various device components for each running application. Based on this profile, an energy efficient content delivery adaptation is performed for the current application.

The simulation testing of DEAS was conducted in different situations in comparison with other solutions. The scenarios involved including WiFi and LTE based heterogeneous wireless network environments, respectively. Tests involving different traffic types and size clearly show the benefits of adopting DEAS over the other solutions.

Device Differentiated Energy Efficient Cooperative Adaptive Multimedia Delivery - EDCAM

EDCAM is proposed as a novel *Energy-efficient Device-differentiated Cooperative Adaptive Multimedia Delivery solution* for heterogeneous wireless networks. EDCAM considers both *interest in content* and *device required quality level* in its *adaptive co-operative delivery* process in order to save energy and maintain high user perceived quality levels. Its major contributions include: an automatic application-aware device profiling process for the calculation of devices' energy constraints; an energy saving two-party cooperative downloading scheme that reduces group management overhead and utilises the multi-home capability of smart devices; a DEAS[148]-based energy efficient quality adaptation algorithm that adapts video quality for energy saving and high QoS in the ever changing device usage environment. Extensive tests have demonstrated how EDCAM outperforms three state of the art solutions resulting in significant improvements in terms of both energy efficiency and user perceived quality levels. For instance, EDCAM demonstrated the energy saving ranging from 15 to 100 percent in comparison with other schemes in all the testing scenarios while maintained higher PSNR score by at least 10 dB.

7.3 Future Work

This section discusses possible future works in the area of energy and quality aware multimedia delivery to mobile devices.

Energy Modelling for Mobile Devices

Although the proposed energy modelling technique has solved some issues which the existing solutions failed to address, there are still much work to be done as new unsolved problems from our field tests was discovered. It is expected that the non linearity of the relation between system utilisation and the actual system power is to be considered by more advanced machine learning models. Only more real tests with different training models on vast range of devices, systems could approach the optimal solution. Apple mobile devices have huge consumers, yet it is more difficult to build a prototype to test on than Android devices. However, this is important work in that Apple mobile devices have substantial market share. An easy and inexpensive yet more accurate state-based WiFi and cellular NIC model is expected as well. There is state machine based model that implements state-based model for WiFi NIC. However their implementation is complicated and results in higer overhead. On the contrary, the dependency of the proposed model on the knowledge of individual device should be reduced as much as possible.

Energy Efficient Adaptive Routing Algorithm

Although the scenario where mobile devices-based ad hoc could serve well in the public area where AP is not available or congested, this type of network is normally deployed in scenarios such as emergency recovery, industry, private or confidential areas and etc. A real test requires AWERA to be configured for such scenarios in order to show how AWERA empowers smart mobile devices by organising them in ad hoc fashion.

Cooperative Adaptive Multimedia Delivery

It is expected to implement EDCAM on real devices in order to fully explore its performance in reality. This is because the prototype of ASP is successful in that it provides plentiful valuable information on the proposed energy modelling technique. Some new observations overlooked in simulations are discovered from the real tests. It is obvious that real tests are superior than simulation tests in demonstrating real performance when dealing with the unexpected or not yet well understood scenarios. Currently only energy constraint and the interest of content are considered in sharing partner selection. The neighbouring device' mobility is missing. The research on mobility brings mobility pattern to the partner selection, which would result in more sensible and optimal selection when multiple users sharing the same interest of content are at the present. Last the mobility and interest of content is valuable information from the view point of content provider. This results in the subject of cloud end optimisation. This subject can well be studied by content providers for cheaper multimedia delivery at higher quality levels.

Bibliography

[1] H. Richmond, "The growth of mobile marketing and tagging," March 2011.

[2] K. Collins, S. Mangold, and G.-M. Muntean, "Supporting mobile devices with wireless LAN/MAN in large controlled environments," *Communications Magazine, IEEE*, vol. 48, pp. 36 –43, Dec 2010.

[3] "Cisco visual networking index: Global mobile data traffic forecast update, 2012 2017," tech. rep., Cisco System, Inc, Feb. 2012.

[4] C. Ge, Z. Sun, and N. Wang, "A survey of power-saving techniques on data centers and content delivery networks," *Communications Surveys Tutorials, IEEE*, vol. 15, no. 3, pp. 1334–1354, 2013.

[5] M. Kennedy, A. Ksentini, Y. Hadjadj-Aoul, and G. Muntean, "Adaptive energy optimization in multimedia-centric wireless devices: A survey," *Communications Surveys Tutorials, IEEE*, vol. 15, no. 2, pp. 768–786, 2013.

[6] "Cisco visual networking index: Global mobile data traffic forecast update, 2011 2016," tech. rep., Cisco System, Inc, Feb. 2012.

[7] G. Boccaletti, M. Loffler, and J. Oppenheim, "How IT can cut carbon emissions," tech. rep., McKinsey and Company, October 2008.

[8] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *Proceedings of the 2010 USENIX conference on USENIX annual technical*

conference, USENIXATC'10, (Berkeley, CA, USA), pp. 21–21, USENIX Association, 2010.

[9] G. Perrucci, F. Fitzek, and J. Widmer, "Survey on energy consumption entities on the smartphone platform," in *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pp. 1 –6, May 2011.

[10] Y. Zhou, T. Song, and X. Wang, "Eabf: Energy efficient self-adaptive bloom filter for network packet processing," in *Communications (ICC), 2012 IEEE International Conference on*, pp. 2729–2734, 2012.

[11] T. Molloy, X. Zheng, O. Ormond, F. Guo, and X. Wang, "Power consumption in a zfilter publish/subscribe based forwarding node," in *Information and Communications Technologies (IETICT 2013), IET International Conference on*, pp. 14–20, 2013.

[12] S. Singh, M. Woo, and C. S. Raghavendra, "Power-aware routing in mobile ad hoc networks," in *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pp. 181–190, 1998.

[13] C.-K. Toh, "Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks," vol. 39, no. 6, pp. 138–147, 2001.

[14] A. S. Arezoomand and M. Pourmina, "Prolonging network operation lifetime with new maximum battery capacity routing in wireless mesh network," in *Proc. 2nd Int Computer and Automation Engineering (ICCAE) Conf*, vol. 4, pp. 319–323, 2010.

[15] R. C. Shah and J. M. Rabaey, "Energy aware routing for low energy ad hoc sensor networksenergy-efficient wake-up scheduling for data collection and aggregation," in *Proc. WCNC2002 Wireless Communications and Networking Conf. 2002 IEEE*, vol. 1, pp. 350–355, 2002.

[16] B. Chen, K. Jamieson, R. Morris, and H. Balakrishnan, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," in *Proceeding of the 7th annual ACMinternational conference on Mobile computing and networking*, 2001.

[17] Y. Wei, C. Wang, M. Song, Y. Ma, and X. Wang, "Energy saving dynamic relaying scheme in wireless cooperative networks using markov decision process," in *Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th*, pp. 1–5, 2012.

[18] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, pp. 660–670, 2002.

[19] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8 - Volume 8*, 2000.

[20] S. Lindsey, C. Raghavendra, and K. M. Sivalingam, "Data gathering algorithms in sensor networks using energy metrics," vol. 13, no. 9, pp. 924–935, 2002.

[21] S. Lindsey and C. S. Raghavendra, "Pegasis: Power-efficient gathering in sensor information systems," in *Proc. IEEE Aerospace*, vol. 3, pp. 3–1125, 2002.

[22] Y. Liu, N. Xiong, Y. Zhao, A. V. Vasilakos, J. Gao, and Y. Jia, "Multi-layer clustering routing algorithm for wireless vehicular sensor networks," 2010. Communications, IET.

[23] S. Ito and K. Yoshigoe, "Consumed-energy-type-aware routing for wireless sensor networks," in *Proc. Wireless Telecommunications Symp. WTS 2007*, pp. 1–6, 2007.

[24] R. Ding and G.-M. Muntean, "An energy-oriented node characteristics-aware routing algorithm for wireless LAN," in *Broadband Multimedia Systems and Broadcasting (BMSB), 2011 IEEE International Symposium on*, pp. 1 –6, Jun 2011.

[25] Y. Yu, R. Govindan, and D. Estrin, "Geographical and energy aware routing: a recursive data dissemination protocol for wireless sensor networks," *UCLA Computer Science Department Technical Report*, 2001.

[26] V. Rodoplu and T. H. Meng, "Minimum energy mobile wireless networks," *IEEE Journal on Selected Areas in Communications*, 1998.

[27] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proceedings of the 7th annual international conference on Mobile computing and networking*, 2001.

[28] I. Stojmenovic and X. Lin, "Power-aware localized routing in wireless networks," in *Proc. 14th Int. Parallel and Distributed Processing Symp. IPDPS 2000*, pp. 371–376, 2000.

[29] H. Zhang and H. Shen, "Energy-efficient beaconless geographic routing in wireless sensor networks," vol. 21, no. 6, pp. 881–896, 2010.

[30] Y. Cao, Z. Sun, and M. Riaz, "Reach-and-spread: A historical geographic routing for delay/disruption tolerant networks," *Networks, IET*, vol. 1, no. 3, pp. 163–170, 2012.

[31] Z. Luo, Z. Sun, and H. Cruickshank, "A delay constrained routing algorithm for LEO satellite networks," in *Proceedings 14th IST Mobile and Wireless Communications Summit*, June 2005.

[32] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Personal Communications*, vol. 7, no. 5, pp. 16–27, 2000.

[33] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, J. Stankovic, and T. Abdelzaher, “Real-time power-aware routing in sensor networks,” in *Proc. 14th IEEE Int. Workshop Quality of Service IWQoS 2006*, pp. 83–92, 2006.

[34] M. EffatParvar, A. Dareshorzadeh, M. Dehghan, and M. R. EffatParvar, “Quality of service support and local recovery for odmrp multicast routing in ad hoc networks,” in *Proc. 4th Int. Conf. Innovations in Information Technology IIT ’07*, pp. 695–699, 2007.

[35] T. Taleb, K. Kashibuchi, A. Leonardi, S. Palazzo, K. Hashimoto, N. Kato, and Y. Nemoto, “A cross-layer approach for an efficient delivery of TCP/RTP-based multimedia applications in heterogeneous wireless networks,” vol. 57, no. 6, pp. 3801–3814, 2008.

[36] K.-Y. Shin, J. Song, J. Kim, M. Yu, and P. S. Mah, “REAR: Reliable energy aware routing protocol for wireless sensor networks,” in *Proc. 9th Int Advanced Communication Technology Conf*, vol. 1, pp. 525–530, 2007.

[37] N. Ouferhat and A. Mellouk, “Energy and delay efficient state dependent routing algorithm in wireless sensor networks,” in *Proc. IEEE 34th Conf. Local Computer Networks LCN 2009*, pp. 1069–1076, 2009.

[38] C. Shanti and A. Sahoo, “Dgram: A delay guaranteed routing and mac protocol for wireless sensornetworks,” vol. 9, no. 10, pp. 1407–1423, 2010.

[39] M. Tariq, Y. P. Kim, J. H. Kim, Y. J. Park, and E. H. Jung, “Energy efficient and reliable routing scheme for wireless sensor networks,” in *Proc. Int. Conf. Communication Software and Networks ICCSN ’09*, pp. 181–185, 2009.

[40] H. Miura and M. Yamamoto, “Content routing with network support using passive measurement in content distribution networks,” in *Proc. Eleventh Int Computer Communications and Networks Conf*, pp. 96–101, 2002.

[41] J. Ni, D. H. K. Tsang, I. S. H. Yeung, and X. Hei, "Hierarchical content routing in large-scale multimedia content delivery network," in *Proc. IEEE Int. Conf. Communications ICC '03*, vol. 2, pp. 854–859, 2003.

[42] A. Mitra, M. Maheswaran, and J. A. Rueda, "Wide-area content-based routing mechanism," in *Proc. Int. Parallel and Distributed Processing Symp*, 2003.

[43] A. Carzaniga, M. J. Rutherford, and A. L. Wolf, "A routing scheme for content-based networking," in *Proc. INFOCOM 2004. Twenty-third AnnualJoint Conf. of the IEEE Computer and Communications Societies*, vol. 2, pp. 918–928, 2004.

[44] W. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pp. 174–185, 1999.

[45] F. Silva, J. Heidemann, R. Govindan, and D. Estrin, *Frontiers in Distributed Sensor Networks*, ch. Directed Diffusion. Boca Raton, Florida, USA: CRC Press, Inc., October 2003.

[46] I. Chlamtac, "Mobile ad hoc networking: imperatives and challenges," *Ad Hoc Networks*, vol. 1, no. 1, pp. 13–64, 2003.

[47] W. Stark, H. Wang, A. Worthen, S. Lafortune, and D. Teneketzis, "Low-energy wireless communication network design," *Wireless Communications, IEEE*, vol. 9, pp. 60 – 72, aug. 2002.

[48] V. Kawadia and P. R. Kumar, "A cautionary perspective on cross-layer design," *Wireless Communications, IEEE [see also IEEE Personal Communications]*, vol. 12, no. 1, pp. 3–11, 2005.

[49] S. Khan, Y. Peng, E. Steinbach, M. Sgroi, and W. Kellerer, “Application-driven cross-layer optimization for video streaming over wireless networks,” *Communications Magazine, IEEE*, vol. 44, pp. 122 – 130, jan. 2006.

[50] L.-U. Choi, W. Kellerer, and E. Steinbach, “Cross layer optimization for wireless multi-user video streaming,” in *Image Processing, 2004. ICIP ’04. 2004 International Conference on*, vol. 3, pp. 2047 – 2050 Vol. 3, oct. 2004.

[51] M. van der Schaar, S. Krishnamachari, S. Choi, and X. Xu, “Adaptive cross-layer protection strategies for robust scalable video transmission over 802.11 wlans,” *Selected Areas in Communications, IEEE Journal on*, vol. 21, pp. 1752 – 1763, dec. 2003.

[52] R. Kravets and P. Krishnan, “Application-driven power management for mobile communication,” *Wirel. Netw.*, vol. 6, pp. 263–277, July 2000.

[53] F. Liu, C. Tsui, and Y. Zhang, “Joint routing and sleep scheduling for lifetime maximization of wireless sensor networks,” vol. 9, no. 7, pp. 2258–2267, 2010.

[54] M. Alicherry, R. Bhatia, and L. Li, “Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks,” in *Proceedings of the 11th annual international conference on Mobile computing and networking*, pp. 58–72, 2005.

[55] S. Saha, M. Hoque, and A. Lukyanenko, “Analyzing energy efficiency of a co-operative content distribution technique,” in *Global Telecommunications Conference, 2011 IEEE*, pp. 1 –6, Dec. 2011.

[56] M.-H. Chen, C.-Y. Yang, C.-Y. Chang, M.-Y. Hsu, K.-H. Lee, and C.-F. Chou, “Towards energy-efficient streaming system for mobile hotspots,” in *Proceedings of the ACM SIGCOMM 2011 conference*, SIGCOMM ’11, (New York, NY, USA), pp. 450 –451, ACM, 2011.

[57] X. Liu, G. Cheung, and C. Chuah, "Structured network coding and cooperative wireless ad-hoc peer-to-peer repair for WWAN video broadcast," *Multimedia, IEEE Transactions on*, vol. 11, pp. 730 –741, june 2009.

[58] H. Hsieh and R. Sivakumar, "On using peer-to-peer communication in cellular wireless data networks," *Mobile Computing, IEEE Transactions on*, vol. 3, no. 1, pp. 57–72, 2004.

[59] M. Stiemerling and S. Kiesel, "Cooperative p2p video streaming for mobile peers," in *Computer Communications and Networks (ICCCN), 2010 Proceedings of 19th International Conference on*, pp. 1–7, 2010.

[60] A. Sharma, V. Navda, R. Ramjee, V. N. Padmanabhan, and E. M. Belding, "Cool-tether: energy efficient on-the-fly wifi hot-spots using mobile phones," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, CoNEXT '09, (New York, NY, USA), pp. 109–120, ACM, 2009.

[61] C.-F. Chiasserini and R. Rao, "A model for battery pulsed discharge with recovery effect," in *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*, pp. 636 –639 vol.2, 1999.

[62] C.-F. Chiasserini and R. Rao, "Improving battery performance by using traffic shaping techniques," *Selected Areas in Communications, IEEE Journal on*, vol. 19, pp. 1385 –1394, jul 2001.

[63] M. Rosu, C. Olsen, C. Narayanaswami, and L. Luo, "PAWP: a power aware web proxy for wireless lan clients," in *Sixth IEEE Workshop on Mobile Computing Systems and Applications*, pp. 206 – 215, 2004.

[64] F. Dogar and P. Steenkiste, "Catnap: Exploiting high bandwidth wireless interfaces to save energy for mobile devices," in *Proc. Int. Conf. Mobile Systems, Applications and Services (MobiSys)*, 2010.

[65] M. C. Rosu, C. M. Olsen, L. Luo, and C. Narayanaswami, "The power-aware streaming proxy architecture," in *First International Workshop on Broadband Wireless Multimedia: Algorithms, Architectures and Applications (BroadWim '04).*, 2004.

[66] S. Chandra and A. Vahdat, "Application-specific network management for energy-aware streaming of popular multimedia formats," in *Proceedings of the General Track of the annual conference on USENIX Annual Technical Conference*, pp. 329–342, 2002.

[67] P. Shenoy, P. Radkovdepartment, and C. Science, "Proxy-assisted power-friendly streaming to mobile devices," in *In MMCN*, pp. 177–191, 2003.

[68] M. Gundlach, S. Doster, H. Yan, D. Lowenthal, S. Watterson, and S. Chandra, "Dynamic, power-aware scheduling for mobile clients using a transparent proxy," in *Parallel Processing, 2004. ICPP 2004. International Conference on*, pp. 557 – 565 vol.1, aug. 2004.

[69] S. A. Akella, R. K. Balan, and N. Bansal, "Protocols for low-power," tech. rep., Carnegie-Mellon University, 2001.

[70] S. Ergen and P. Varaiya, "Pedamacs: power efficient and delay aware medium access protocol for sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 5, pp. 920 – 930, july 2006.

[71] P. Suriyachai, U. Roedig, and A. Scott, "Implementation of a mac protocol for qos support in wireless sensor networks," in *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*, pp. 1 –6, march 2009.

[72] G. Lu, B. Krishnamachari, and C. Raghavendra, "An adaptive energy-efficient and low-latency mac for data gathering in wireless sensor networks," in *Parallel*

and Distributed Processing Symposium, 2004. Proceedings. 18th International, p. 224, april 2004.

[73] M. Sichitiu, "Cross-layer scheduling for power efficiency in wireless sensor networks," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 1740 –1750 vol.3, march 2004.

[74] Y. Wu, X.-Y. Li, Y. Liu, and W. Lou, "Energy-efficient wake-up scheduling for data collection and aggregation," vol. 21, no. 2, pp. 275–287, 2010.

[75] A. M. Bernardos, P. Tarrio, and J. R. Casar, "An energy aware routing algorithm for ad hoc and sensor networks: Concept and performance," in *Proc. IEEE 17th Int Personal, Indoor and Mobile Radio Communications Symp*, pp. 1–5, 2006.

[76] J. Chen, T. Lv, and H. Zheng, "Cross-layer design for QoS wireless communications," in *Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on*, vol. 2, pp. II – 217–20 Vol.2, May 2004.

[77] Y. Liu, I. Elhanany, and H. Qi, "An energy-efficient QoS-aware media access control protocol for wireless sensor networks," in *Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on*, pp. 3 pp. –191, nov. 2005.

[78] N. Saxena, A. Roy, and J. Shin, "Dynamic duty cycle and adaptive contention window based qos-mac protocol for wireless multimedia sensor networks," *Comput. Netw.*, vol. 52, pp. 2532–2542, September 2008.

[79] R. S. Tupelly and J. Zhang, "Opportunistic scheduling for streaming video in wireless networks," in *Hopkins University*, 2003.

[80] Q. Liu, S. Zhou, and G. Giannakis, "Cross-layer scheduling with prescribed qos guarantees in adaptive wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 23, pp. 1056 – 1066, may 2005.

[81] W. L. Bircher and L. K. John, “Analysis of dynamic power management on multi-core processors,” in *Proceedings of the 22nd annual international conference on Supercomputing*, pp. 327–338, 2008.

[82] A. Mahesri and V. Vardhan, “Power consumption breakdown on a modern laptop,” pp. 165–180, 2005.

[83] D. C. Snowdon, E. L. S., S. M. Petters, and G. Heiser, “Koala: a platform for os-level power management,” in *Proceedings of the 4th ACM European conference on Computer systems*, pp. 289–302, 2009.

[84] G. Chinn, S. Desai, E. DiStefano, K. Ravichandran, and S. Thakkar, “Mobile PC platforms enabled with intel centrino mobile technology,” *Intel Technology Journal*, vol. 7, May 2003.

[85] “IEEE std. 802.11a: Wireless LAN medium access control (MAC) and physical layer (PHY) specification,” 1997.

[86] C. Eklund, K. Stanwood, S. Wang, and E. C. Inc, “IEEE standard 802.16: A technical overview of the wireless MAN air interface for broadband wireless access,” *IEEE Communications Magazine*, vol. 40, pp. 98–107, 2002.

[87] T. Brady, D. Bodas, D. Gabel, B. Griffith, T. Niemela, and D. Perchlik, “Pc power consumption - a challenge and opportunity,” in *Electronics and the Environment, 2005. Proceedings of the 2005 IEEE International Symposium on*, pp. 140–145, 2005.

[88] J. Charles, P. Jassi, N. Ananth, A. Sadat, and A. Fedorova, “Evaluation of the intel core i7 turbo boost feature,” in *Workload Characterization, 2009. IISWC 2009. IEEE International Symposium on*, pp. 188–197, 2009.

[89] J. M. Cebrian, L. Natvig, and J. C. Meyer, “Improving energy efficiency through parallelization and vectorization on intel core i5 and i7 processors,” in *High*

Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion:, pp. 675–684, 2012.

[90] D. Sadowy, "New microprocessor features for lower power and increased performance: doing "moore's" with less," in *Electronics and the Environment, 2005. Proceedings of the 2005 IEEE International Symposium on*, pp. 146–150, 2005.

[91] J. Khan, S. Bilavarn, and C. Belleudy, "Energy analysis of a dvfs based power strategy on arm platforms," in *Faible Tension Faible Consommation (FTFC), 2012 IEEE*, pp. 1 –4, june 2012.

[92] D. Bull, S. Das, K. Shivashankar, G. Dasika, K. Flautner, and D. Blaauw, "A power-efficient 32 bit arm processor using timing-error detection and correction for transient-error tolerance and adaptation to pvt variation," *Solid-State Circuits, IEEE Journal of*, vol. 46, pp. 18 –31, jan. 2011.

[93] S. Das, C. Tokunaga, S. Pant, W.-H. Ma, S. Kalaiselvan, K. Lai, D. Bull, and D. Blaauw, "Razorii: In situ error detection and correction for pvt and ser tolerance," *Solid-State Circuits, IEEE Journal of*, vol. 44, pp. 32 –48, jan. 2009.

[94] K. Bowman, J. Tschanz, N. S. Kim, J. Lee, C. Wilkerson, S.-L. Lu, T. Karnik, and V. De, "Energy-efficient and metastability-immune timing-error detection and instruction-replay-based recovery circuits for dynamic-variation tolerance," in *Solid-State Circuits Conference, 2008. ISSCC 2008. Digest of Technical Papers. IEEE International*, pp. 402 –623, feb. 2008.

[95] J. Tschanz, K. Bowman, S.-L. Lu, P. Aseron, M. Khellah, A. Raychowdhury, B. Geuskens, C. Tokunaga, C. Wilkerson, T. Karnik, and V. De, "A 45nm resilient and adaptive microprocessor core for dynamic variation tolerance," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, pp. 282 –283, feb. 2010.

[96] J. Tschanz, K. Bowman, S. Walstra, M. Agostinelli, T. Karnik, and V. De, "Tunable replica circuits and adaptive voltage-frequency techniques for dynamic voltage, temperature, and aging variation tolerance," in *VLSI Circuits, 2009 Symposium on*, pp. 112 –113, june 2009.

[97] M. Dong and L. Zhong, "Power modeling and optimization for OLED displays," *Mobile Computing, IEEE Transactions on*, vol. 11, pp. 1587 –1599, sept. 2012.

[98] C. Lin, P. Hsiu, and C. Hsieh, "Dynamic backlight scaling optimization: A cloud-based energy-saving service for mobile streaming applications," *IEEE Transactions on Computers*, vol. 99, no. PrePrints, p. 1, 2012.

[99] J. Flinn and M. Satyanarayanan, "Energy-aware adaptation for mobile applications," in *Proceedings of the seventeenth ACM symposium on Operating systems principles*, SOSP '99, (New York, NY, USA), pp. 48–63, ACM, 1999.

[100] P. Ranganathan, E. Geelhoed, M. Manahan, and K. Nicholas, "Energy-aware user interfaces and energy-adaptive displays," *Computer*, vol. 39, pp. 31 – 38, march 2006.

[101] D. Shin, Y. Kim, N. Chang, and M. Pedram, "Dynamic voltage scaling of oled displays," in *Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE*, pp. 53 –58, june 2011.

[102] F. Maker and Y. Chan, "A survey on android v.s. linux," *Management*, pp. 1–10, 2009.

[103] C. Margi, K. Obraczka, and R. Manduchi, "Characterizing system level energy consumption in mobile computing platforms," in *Wireless Networks, Communications and Mobile Computing, 2005 International Conference on*, vol. 2, pp. 1142–1147 vol.2, 2005.

[104] P. Ljung, "Opportunities for energy savings in mobile devices," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2011 IEEE 22nd International Symposium on*, pp. 2394–2401, 2011.

[105] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, and L. Yang, "Accurate online power estimation and automatic battery behavior based power model generation for smartphones," in *Proceedings of the 8th IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, CODES/ISSS '10, (New York, NY, USA), pp. 105–114, ACM, 2010.

[106] M. Dong and L. Zhong, "Self-constructive high-rate system energy modeling for battery-powered mobile systems," in *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, MobiSys '11, (New York, NY, USA), pp. 335–348, ACM, 2011.

[107] A. Pathak, Y. C. Hu, M. Zhang, P. Bahl, and Y.-M. Wang, "Fine-grained power modeling for smartphones using system call tracing," in *Proceedings of the 6th Conference on Computer Systems*, EuroSys '11, (New York, NY, USA), pp. 153–168, ACM, 2011.

[108] A. Pathak, Y. C. Hu, and M. Zhang, "Where is the energy spent inside my app ? : fine grained energy accounting on smartphones with eprof," in *Proceedings of the 7th ACM European Conference on Computer Systems*, EuroSys '12, (New York, NY, USA), pp. 29–42, ACM, 2012.

[109] C. Wang and Z. Li, "A computation offloading scheme on handheld devices," *J. Parallel Distrib. Comput.*, vol. 64, pp. 740–746, 2004.

[110] S. Gitzenis and N. Bambos, "Joint task migration and power management in wireless computing," *Mobile Computing, IEEE Transactions on*, vol. 8, pp. 1189 –1204, sept. 2009.

[111] U. Kremer, J. Hicks, and J. Rehg, "A compilation framework for power and energy management on mobile computers," in *In International Workshop on Languages and Compilers for Parallel Computing*, 2001.

[112] M. Burrows, D. J. Wheeler, M. Burrows, and D. J. Wheeler, "A block-sorting lossless data compression algorithm," tech. rep., HP Labs, 1994.

[113] T. A. Welch, "A technique for high-performance data compression," *Computer*, vol. 17, pp. 8–19, June 1984.

[114] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Transactions on Information Theory*, vol. 23, no. 3, pp. 337–343, 1977.

[115] B. R. Iyer and D. Wilhite, "Data compression support in databases," in *Proceedings of the 20th International Conference on Very Large Databases*, pp. 695–704, 1994.

[116] W. Ng and C. Ravishankar, "Block-oriented compression techniques for large statistical databases," *IEEE Transactions on Knowledge and Data Engineering*, vol. 9, pp. 314–328, 1997.

[117] D. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the IRE*, vol. 40, pp. 1098 –1101, sept. 1952.

[118] G. Langdon, "Arithmetic coding," *IBM J. Res. Develop*, vol. 23, pp. 149–162, 1979.

[119] S. Golomb, "Run-length encodings (Corresp.)," *Information Theory, IEEE Transactions on*, vol. 12, no. 3, pp. 399–401, 1966.

[120] N. Ahmed, T. Nataraja, and K. Rao, "Discrete cosine transform," *IEEE transactions on computers*, vol. C 23, no. 1, pp. 90–93, 1974.

[121] Y. Fisher, ed., *Fractal image compression: theory and application*. London, UK: Springer-Verlag, 1995.

[122] C. C. Cutler, "Differential quantization of communication signals," July 1952.

[123] F. de Jager, "Delta modulation: a method of pcm transmission using the one unit code," tech. rep., Phillips Research, 1952.

[124] S. Mccanne, M. Vetterli, and V. Jacobson, "Low-complexity video coding for receiver-driven layered multicast," *IEEE Journal on Selected Areas in Communications*, vol. 15, pp. 983–1001, 1997.

[125] E. Masala, D. Quaglia, and J. De Martin, "Adaptive picture slicing for distortion-based classification of video packets," in *Multimedia Signal Processing, 2001 IEEE Fourth Workshop on*, pp. 111 –116, 2001.

[126] J. Shin, J. W. Kim, and C.-C. Kuo, "Quality-of-service mapping mechanism for packet video in differentiated services network," *Multimedia, IEEE Transactions on*, vol. 3, pp. 219 –231, jun 2001.

[127] M. Schroeder and B. Atal, "Code-excited linear prediction(celp): High-quality speech at very low bit rates," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '85.*, vol. 10, pp. 937 – 940, apr 1985.

[128] C. Im and S. Ha, "Energy optimization for latency- and quality-constrained video applications," *IEEE Des. Test*, vol. 21, pp. 358–366, September 2004.

[129] "H.261:video codec for audio visual services at p x 64 kbit/s," 1993.

[130] "ISO/IEC Std. 10 918-1: Information technology digital compression and coding of continuous-tone still images - requirements and guidelines," 1994.

[131] C.-F. Chiasserini and E. Magli, "Energy consumption and image quality in wireless video-surveillance networks," in *Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on*, vol. 5, pp. 2357 – 2361 vol.5, sept. 2002.

[132] "ISO/IEC Std.11 172: MPEG-1 coding of moving pictures and associated audio for digital storage media up to 1.5 Mbits/s," 1993.

[133] "ISO/IEC Std.13 818: MPEG-2 generic coding of moving pictures and associated audio information," 1994.

[134] "ISO/IEC Std. 14496-2 MPEG-4 visual," 2004.

[135] "ISO/IEC Std 14496-10 MPEG-4 AVC," 2005.

[136] "ISO/IEC Std.14 496: Information technology - coding of audio-visual objects," 2001.

[137] "ISO/IEC Std.15 938: Information technology multimedia content description interface," 2002.

[138] S. F. Chang, T. Sikora, and A. Puri, "Overview of the MPEG-7 standard," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 11, pp. 688–695, June 2001.

[139] K. C., K. K., and M. Pedram, "Energy-aware MPEG-4 FGS streaming," in *Design Automation Conference, 2003. Proceedings*, pp. 912 – 915, june 2003.

[140] G.-M. Muntean, P. Perry, and L. Murphy, "A new adaptive multimedia streaming system for all-ip multi-service networks," *Broadcasting, IEEE Transactions on*, vol. 50, pp. 1 – 10, march 2004.

[141] C. N. Taylor, S. Dey, and D. Panigrahi, "Energy/latency/image quality tradeoffs in enabling mobile multimedia communication," in *in Enabling Mobile Multimedia Communications, Software Radio: Technologies and Services 2001*, pp. 55–66, Springer Verlag, 2001.

[142] H. Liu and M. El Zarki, "Adaptive source rate control for real-time wireless video transmission," *Mob. Netw. Appl.*, vol. 3, pp. 49–60, June 1998.

[143] A. I., K. C., and M. P., "Energy-aware wireless video streaming.," in *ESTImedia* (G. Fohler and R. Marculescu, eds.), pp. 48–55, 2003.

[144] L. Kaddar, Y. Hadjadj-Aoul, and A. Mehaoua, "Evan: Energy-aware svc video streaming over wireless ad hoc networks," in *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pp. 1–5, 2011.

[145] L. Kaddar and A. Mehaoua, "Estrel: Transmission and reception energy saving model for wireless ad hoc networks," in *Local Computer Networks, 2007. LCN 2007. 32nd IEEE Conference on*, pp. 784–785, 2007.

[146] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the h.264/avc standard," *IEEE Trans. Cir. and Sys. for Video Technol.*, vol. 17, pp. 1103–1120, Sept. 2007.

[147] R. Trestian, O. Ormond, and G.-M. Muntean, "Signal strength-based adaptive multimedia delivery mechanism," in *Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on*, pp. 297 –300, Oct. 2009.

[148] R. Ding and G.-M. Muntean, "Device characteristics-based differentiated energy-efficient adaptive solution for video delivery over heterogeneous wireless networks," in *Wireless Communications and Networking Conference, 2013. WCNC. 2013 IEEE*, Apr. 2013.

[149] A.-M. Alt and D. Simon, "Control strategies for h.264 video decoding under resources constraints," *SIGOPS Oper. Syst. Rev.*, vol. 44, pp. 53–58, Aug. 2010.

[150] J. S. Park and T. Ogunfunmi, "A VLSI architecture of svc encoder for mobile system," in *Consumer Electronics (ICCE), 2011 IEEE International Conference on*, pp. 809–810, 2011.

[151] J. Flinn and M. Satyanarayanan, "Energy-aware adaptation for mobile applications," in *Proceedings of the seventeenth ACM symposium on Operating systems principles*, pp. 48–63, 1999.

[152] L. Rizzo and L. Vicisano, "Replacement policies for a proxy cache," *Networking, IEEE/ACM Transactions on*, vol. 8, pp. 158 –170, apr 2000.

[153] S. Michel, K. Nguyen, A. Rosenstein, L. Zhang, S. Floyd, and V. Jacobson, "Adaptive web caching: towards a new global caching architecture," *Comput. Netw. ISDN Syst.*, vol. 30, pp. 2169–2177, November 1998.

[154] R. Rejaie, H. Yu, M. Handley, and D. Estrin, "Multimedia proxy caching mechanism for quality adaptive streaming applications in the internet," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, pp. 980 –989 vol.2, 2000.

[155] S. Sen, J. Rexford, and D. Towsley, "Proxy prefix caching for multimedia streams," in *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, pp. 1310 –1319 vol.3, mar 1999.

[156] R. Rejaie, H. Yu, M. Handley, and D. Estrin, "Proxy caching mechanism for multimedia playback streams in the internet," in *in Proceedings of the 4th International Web Caching Workshop*, 1999.

[157] S. Chen, H. Wang, B. Shen, S. Wee, and X. Zhang, "Segment-based proxy caching for internet streaming media delivery," in *in IEEE Multimedia*, pp. 59–67, 2005.

[158] K.-L. Wu, P. S. Yu, and J. L. Wolf, "Segment-based proxy caching of multimedia streams," in *Proceedings of the 10th international conference on World Wide Web*, pp. 36–44, 2001.

[159] Y. Chae, K. Guo, M. Buddhikot, S. Suri, and E. Zegura, "Silo, rainbow, and caching token: schemes for scalable, fault tolerant stream caching," *Selected Areas in Communications, IEEE Journal on*, vol. 20, pp. 1328 – 1344, sep 2002.

[160] A. Pathak, Y. C. Hu, and M. Zhang, "Where is the energy spent inside my app ? : fine grained energy accounting on smartphones with eprof," in *Proceedings of the 7th ACM European Conference on Computer Systems*, EuroSys '12, (New York, NY, USA), pp. 29–42, ACM, 2012.

[161] R. Ding and G.-M. Muntean, "A context-aware cross-layer energy-efficient adaptive routing algorithm for WLAN communications," in *Local Computer Networks, 2012. LCN 2012. IEEE 37th Conference on*, pp. 176 –179, Oct. 2012.

[162] R. Ding and G.-M. Muntean, "A novel device and application-aware energy efficient routing algorithm for WLANs," in *GLOBECOM Workshops (GC Wkshps), 2012 IEEE*, Dec. 2012.

[163] T. Hastie, R. J. Tibshirani, and J. H. Friedman, *The elements of statistical learning : data mining, inference, and prediction*. Springer series in statistics, New York, N.Y. Springer, 2009.

[164] Q. Li, Q. Meng, J. Cai, H. Yoshino, and A. Mochida, "Applying support vector machine to predict hourly cooling load in the building," *Applied Energy*, vol. 86, no. 10, pp. 2249–2256, 2009.

[165] M. Dong, Y. Choi, and L. Zhong, "Power modeling of graphical user interfaces on oled displays," in *Proceedings of the 46th Annual Design Automation Conference*, DAC '09, (New York, NY, USA), pp. 652–657, ACM, 2009.

[166] R. Murmuria, J. Medsger, A. Stavrou, and J. Voas, "Mobile application and device power usage measurements," in *Software Security and Reliability (SERE), 2012 IEEE Sixth International Conference on*, pp. 147–156, 2012.

[167] L. Wang, A. Ukhanova, and E. Belyaev, "Power consumption analysis of constant bit rate data transmission over 3G mobile wireless networks," in *ITS Telecommunications (ITST), 2011 11th International Conference on*, pp. 217–223, 2011.

[168] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing." RFC 3561, Jul 2003.

[169] R. Trestian, A.-N. Moldovan, O. Ormond, and G.-M. Muntean, "Energy consumption analysis of video streaming to android mobile devices," in *Network Operations and Management Symposium (NOMS), 2012 IEEE*, pp. 444 –452, Apr. 2012.

[170] S. Lee, G.-M. Muntean, and A. Smeaton, "Performance-aware replication of distributed pre-recorded IPTV content," *Broadcasting, IEEE Transactions on*, vol. 55, pp. 516 –526, Jun. 2009.

[171] H. Luo, X. Meng, R. Ramjee, P. Sinha, and L. Li, "The design and evaluation of unified cellular and ad-hoc networks," *Mobile Computing, IEEE Transactions on*, vol. 6, no. 9, pp. 1060–1074, 2007.

[172] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "A close examination of performance and power characteristics of 4g lte networks," in *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, MobiSys '12, (New York, NY, USA), pp. 225–238, ACM, 2012.